

**AD-A233 756**

**Application of a Conjugate Gradient Method  
to the Synthesis of Phase-Only Planar Arrays**

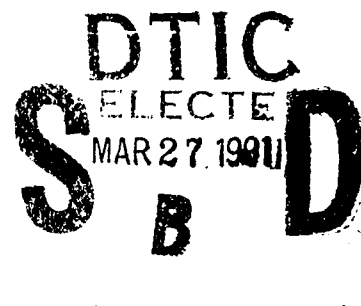
Prepared by

**T. J. PETERS**  
Electronics Research Laboratory  
Laboratory Operations

15 February 1991

Prepared for

**SPACE SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
Los Angeles Air Force Base  
P. O. Box 92960  
Los Angeles, CA 90009-2960**



Engineering and Technology Group

**THE AEROSPACE CORPORATION**  
El Segundo, California

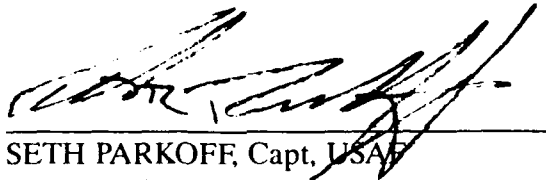
APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED

71 0 0 0 0 8

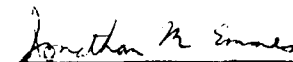
This report was submitted by The Aerospace Corporation, El Segundo, CA 90245, under Contract No. F04/01-88-C-0089 with the Space Systems Division, P. O. Box 92960, Los Angeles, CA 90009-2960. It was reviewed and approved for The Aerospace Corporation by B. K. Janousek, Director, Electronics Research Laboratory. Capt Seth Parkoff was the project officer for the Mission-Oriented Investigation and Experimentation (MOIE) Program.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.



SETH PARKOFF, Capt, USAF  
MOIE Project Officer  
SSD/MHE



JONATHAN M. EMMES, Maj, USAF  
MOIE Program Manager  
AFSTC/WCO CL-AB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				
1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR-0090(5925-05)-1		5. MONITORING ORGANIZATION REPORT NUMBER(S) SSD-TR-91-03		
6a. NAME OF PERFORMING ORGANIZATION The Aerospace Corporation Laboratory Operations		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION Space Systems Division
6c. ADDRESS (City, State, and ZIP Code) El Segundo CA 90245 4591		7b. ADDRESS (City, State, and ZIP Code) Los Angeles Air Force Base Los Angeles, CA 90009-2960		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F04701-88-C-0089
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
		WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Application of a Conjugate Gradient Method to the Synthesis of Phase-Only Planar Arrays				
12. PERSONAL AUTHOR(S) Peters, Timothy J.				
13a. TYPE OF REPORT		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1991 February 15
15. PAGE COUNT 59				
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Antenna Array Synthesis Conjugate Gradient	
FIELD	GROUP	SUB-GROUP		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>The problem of minimizing the peak sidelobe level of a planar array of dipoles using phase-only synthesis is investigated. A hybrid nonlinear function minimization scheme is developed using a Taylor series approximation and a conjugate gradient algorithm. A weighted average peak sidelobe level function is introduced to yield a more stable numerical procedure. The peak sidelobe level is found to decrease in proportion to the logarithm of the aperture behavior.</p>				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL			22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

## CONTENTS

I. INTRODUCTION .....	3
II. GEOMETRY AND FAR FIELD .....	5
III. A NONLINEAR CONJUGATE GRADIENT METHOD .....	9
IV. MINIMIZING MAXIMUM SIDELobe LEVEL .....	15
V. RESULTS .....	23
VI. CONCLUSIONS .....	33
REFERENCES .....	35
APPENDIX: FORTRAN 77 PROGRAM .....	37

## FIGURES

1. Geometry of Planar Array .....	5
2. Weighted Average Minimization .....	16
3. Sampled $u - v$ Plane .....	18
4. Peak Sidelobe Level .....	24
5. Aperture Efficiency .....	25
6. Half-Power Beam Width .....	26
7. Beam Width Between First Nulls .....	27
8. Uniform Phase Power Pattern in Cylindrical Coordinates .....	29
9. Synthesized Phase Power Pattern in Cylindrical Coordinates .....	30
10. Uniform Phase Power Pattern in Spherical Coordinates .....	31
11. Synthesized Phase Power Pattern in Spherical Coordinates .....	32

## I. INTRODUCTION

The conjugate gradient method was developed by Hestenes and independently by Steifel, but the initial comprehensive treatment was a joint publication(ref. 1). The first application of the method to the synthesis of antenna patterns appears to have been implemented by Fong and Birgenheier(ref. 2) to compute the line source required to produce a desired power pattern. This application concerned the solution of a Fredholm equation of the first kind by minimizing an appropriate weighted least-squares integral. The algorithm used in reference 2 is a computationally fast nonlinear minimization scheme, but it has two major disadvantages. First, it may not converge for many functions that represent the power pattern of a discrete array. Second, an external one-dimensional minimization algorithm must be supplied by the user. These two disadvantages can lead to an unpredictable breakdown in the algorithm.

This study has two major parts. The first part is the development of a reliable way to apply a conjugate gradient algorithm to the minimization of an arbitrary nonlinear function. The second part is the application of that algorithm to the minimization of the maximum sidelobe level of a planar array of dipoles using phase-only synthesis.

<b>Accession For</b>	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

## II. GEOMETRY AND FAR FIELD

The array configuration chosen for this study is a two-dimensional array of half-wavelength horizontal dipoles placed above a ground plane. The dipoles are placed such that there is quadrant symmetry, and a circular perimeter is approximated. The top view of a sample array is shown in Fig. 1.

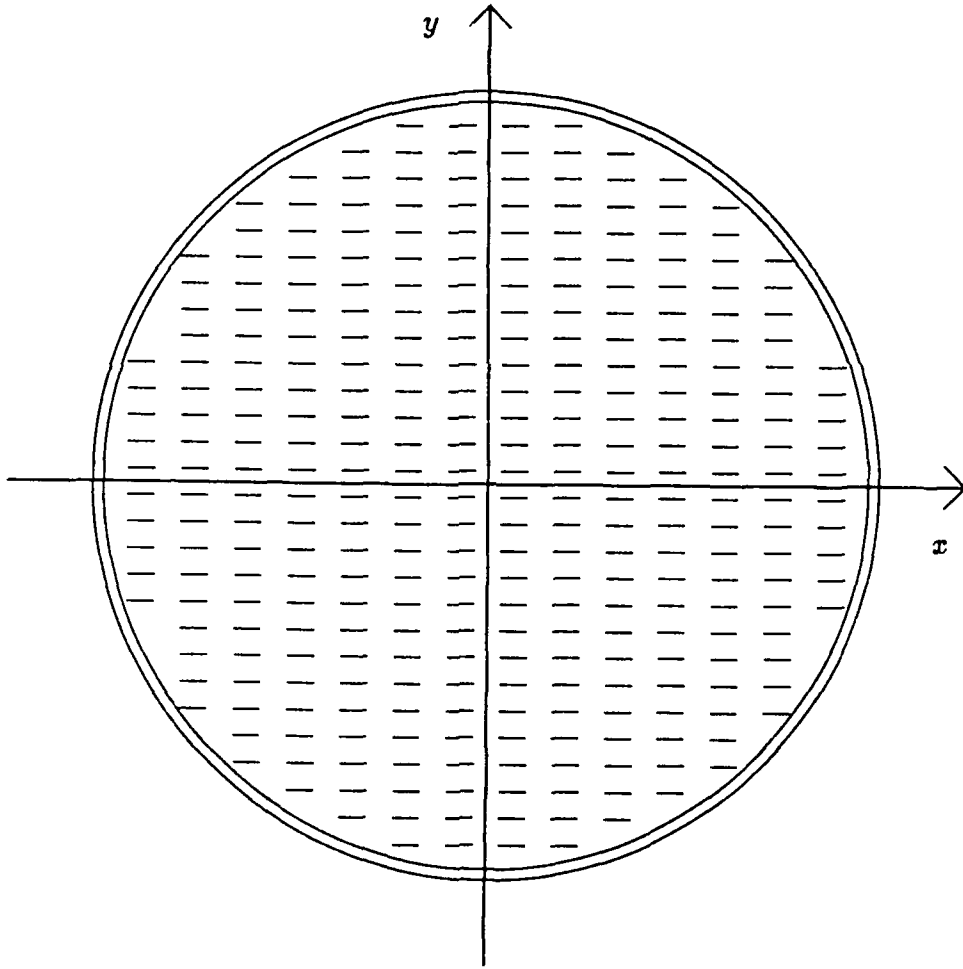


Fig. 1. Geometry of Planar Array

Assuming an  $e^{j\omega t}$  time convention, the total far-zone electric field due to

a single  $x$ -directed current source is given by

$$\vec{E} = j \frac{e^{-jk_0 R}}{4\pi R} \left[ \sin(\phi)\hat{\theta} + \cos(\theta)\cos(\phi)\hat{\phi} \right] N_x \quad (2.1)$$

where

$$N_x = \iiint J_x(x, y, z) e^{j(u x + v y + \sqrt{k_0^2 - u^2 - v^2} z)} dx dy dz \quad (2.2)$$

and

$$u = k_0 \sin(\theta) \cos(\phi) \quad (2.3)$$

$$v = k_0 \sin(\theta) \sin(\phi). \quad (2.4)$$

If the source consists of  $4N$  equal-amplitude, half-wavelength dipoles placed a distance  $h$  above a ground plane and if the  $k$ th current has the form

$$J_x = I_k \delta(y - y_k) \delta(z) \sin \left[ k_0 \left( \frac{\lambda_0}{4} - |x - x_k| \right) \right] \quad |x - x_k| \leq \frac{\lambda_0}{4} \quad (2.5)$$

then equation (2.2) reduces to

$$N_x = j 4 k_0 \sin \left( h \sqrt{k_0^2 - u^2 - v^2} \right) \frac{\cos \left( \frac{\pi}{2 k_0} u \right)}{k_0^2 - u^2} \sum_{k=1}^{4N} I_k e^{j[\gamma_k + \psi_k(u, v)]}. \quad (2.6)$$

The far-zone power pattern is then given by

$$P(\gamma, u, v) = \beta(u, v) \left[ \left( \sum_{k=1}^{4N} I_k \cos[\gamma_k + \psi_k(u, v)] \right)^2 + \left( \sum_{k=1}^{4N} I_k \sin[\gamma_k + \psi_k(u, v)] \right)^2 \right] \quad (2.7)$$

where

$$\beta(u, v) = \frac{1}{k_0^2 - u^2} \sin^2 \left( h \sqrt{k_0^2 - u^2 - v^2} \right) \cos^2 \left( \frac{\pi}{2 k_0} u \right) \quad (2.8)$$

and  $\psi_k(u, v) = u x_k + v y_k$ . Because quadrant symmetry is assumed, the power pattern simplifies to

$$P(\gamma, u, v) = \beta(u, v) \left[ \left( \sum_{k=1}^N I_k \zeta_k(u, v) \cos(\gamma_k) \right)^2 + \left( \sum_{k=1}^N I_k \zeta_k(u, v) \sin(\gamma_k) \right)^2 \right] \quad (2.9)$$

where  $\zeta_k = \cos(u x_k) \cos(v y_k)$ .

Antenna pattern synthesis involves controlling the power pattern. For a known amplitude excitation, control of the power pattern may be achieved by adjusting the relative phase excitation of each element. The goal of the synthesis will be to maximize the power in the main beam while simultaneously minimizing the peak sidelobe power. Observation of equation (2.9) indicates that the far-zone power pattern is a nonlinear function of the unknown excitation phase. Therefore, a nonlinear minimization scheme will be required in the synthesis.

### III. A NONLINEAR CONJUGATE GRADIENT METHOD

As the previous section indicated, minimizing the sidelobe level power of an antenna pattern involves the minimization of a nonlinear function. A conjugate gradient method may be used to find the minimum value of the nonlinear function  $f(\gamma)$ , where

$$\gamma = [\gamma_1 \quad \gamma_2 \quad \cdots \quad \gamma_N]^T \quad (3.1)$$

with each  $\gamma_k$  representing the relative phase shift for the  $k$ th element. It will be assumed that  $f(\gamma)$  is a nonconvex, twice-differentiable real function of a real variable. This implies that there are multiple solutions to the equation

$$\nabla f = 0 \quad (3.2)$$

where  $\nabla f$  is the gradient of  $f(\gamma)$  defined by

$$\nabla f = \left[ \frac{\partial f}{\partial \gamma_1} \quad \frac{\partial f}{\partial \gamma_2} \quad \cdots \quad \frac{\partial f}{\partial \gamma_N} \right]^T. \quad (3.3)$$

Each solution of equation (3.2) is called a "stationary" or "critical" point. A conjugate gradient solution is possible by assuming an initial estimate  $\gamma^1$  and approximating  $f(\gamma)$  as a quadratic truncation of a Taylor series expanded around this point as

$$f(\gamma) = f(\gamma^1) - (\gamma - \gamma^1)^T G + \frac{1}{2}(\gamma - \gamma^1)^T H(\gamma - \gamma^1) + O(|\gamma - \gamma^1|^3) \quad (3.4)$$

where  $G$  is the negative gradient of  $f(\gamma)$  evaluated at  $\gamma^1$  given by

$$G = -(\nabla f)(\gamma^1) = \left[ \frac{\partial f}{\partial \gamma_1}(\gamma^1) \quad \frac{\partial f}{\partial \gamma_2}(\gamma^1) \quad \cdots \quad \frac{\partial f}{\partial \gamma_N}(\gamma^1) \right]^T \quad (3.5)$$

and  $H$  is the Hessian of  $f(\gamma)$  evaluated at  $\gamma^1$  given by

$$H = (\nabla \nabla f)(\gamma^1) = \begin{bmatrix} \frac{\partial^2 f}{\partial \gamma_1^2}(\gamma^1) & \frac{\partial^2 f}{\partial \gamma_2 \partial \gamma_1}(\gamma^1) & \cdots & \frac{\partial^2 f}{\partial \gamma_N \partial \gamma_1}(\gamma^1) \\ \frac{\partial^2 f}{\partial \gamma_1 \partial \gamma_2}(\gamma^1) & \frac{\partial^2 f}{\partial \gamma_2^2}(\gamma^1) & \cdots & \frac{\partial^2 f}{\partial \gamma_N \partial \gamma_2}(\gamma^1) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial \gamma_1 \partial \gamma_N}(\gamma^1) & \frac{\partial^2 f}{\partial \gamma_2 \partial \gamma_N}(\gamma^1) & \cdots & \frac{\partial^2 f}{\partial \gamma_N^2}(\gamma^1) \end{bmatrix}. \quad (3.6)$$

It is important to note that the Hessian matrix  $H$  is always symmetric but not necessarily positive definite. The matrix would be positive definite if, given any nonzero vector  $p$ , the condition  $\langle Hp, p \rangle > 0$  is satisfied. The brackets denote an inner product between two  $N \times 1$  vectors defined by

$$\langle p, q \rangle = \sum_{n=1}^N p_n q_n. \quad (3.7)$$

Making the substitution  $z = \gamma - \gamma^1$  and excluding the constant and higher order terms of equation (3.4) yields the function

$$c(z) = \frac{1}{2} z^T H z - z^T G \quad (3.8)$$

which, when minimized, produces the same solution as solving the linear system of equations  $H z = G$ . Because  $H$  and  $G$  are implicit functions of  $z$ , the algorithm must be restarted periodically to yield the correct solution to the original problem. If  $H$  is positive definite, then a possible algorithm which iterates  $K$  times with  $L$  restarts is given as follows:

*begin loop for  $l=1, \dots, L$*

*compute  $G_l = -(\nabla f)(\gamma^l)$ ,  $H_l = (\nabla \nabla f)(\gamma^l)$  and set  $z^1 = 0$*

$$r^1 = G_l \quad (3.9)$$

$$\beta_0 = \frac{1}{\langle r^1, r^1 \rangle} \quad (3.10)$$

$$p^1 = \beta_0 r^1 \quad (3.11)$$

*begin loop for  $k=1, \dots, K$*

$$\alpha_k = \frac{1}{\langle H_l p^k, p^k \rangle} \quad (3.12)$$

$$z^{k+1} = z^k + \alpha_k p^k \quad (3.13)$$

$$r^{k+1} = r^k - \alpha_k H_l p^k \quad (3.14)$$

$$err = \frac{\langle r^{k+1}, r^{k+1} \rangle}{\langle r^k, r^k \rangle} \quad (3.15)$$

*if err < tolerance, terminate k loop, else*

$$\beta_k = \frac{1}{\langle r^{k+1}, r^{k+1} \rangle} \quad (3.16)$$

$$p^{k+1} = p^k + \beta_k r^{k+1} \quad (3.17)$$

*continue k loop*

$$\gamma^{l+1} = \gamma^l + z^{k+1} \quad (3.18)$$

*continue l loop*

This algorithm requires that  $H$  be explicitly known in order to compute  $\alpha_k$  and  $r^{k+1}$ . An alternate algorithm that does not require computation of  $H$  may be derived by noting that  $r^{k+1} = -(\nabla c)(z^k) \approx -(\nabla f)(\gamma^k)$ . Also,  $\alpha_k$  may be approximated by solving the one-dimensional minimization problem

$$\text{choose } \alpha_k \text{ to minimize } f(\gamma^k + \alpha_k p^k) \quad (3.19)$$

This yields an alternate algorithm proposed in references 2 and 3 which does not require the explicit calculation of  $H$ . The algorithm is given as follows:

*begin loop for l = 1, ..., L*

*compute*  $G_l = -(\nabla f)(\gamma^l)$ ,  $H_l = (\nabla \nabla f)(\gamma^l)$  *and set*  $z^1 = 0$

$$r^1 = G_l \quad (3.20)$$

$$\beta_0 = \frac{1}{\langle r^1, r^1 \rangle} \quad (3.21)$$

$$p^1 = \beta_0 r^1 \quad (3.22)$$

*begin loop for k = 1, ..., K*

$$\text{choose } \alpha_k \text{ to minimize } f(\gamma^k + \alpha_k p^k) \quad (3.23)$$

$$z^{k+1} = z^k + \alpha_k p^k \quad (3.24)$$

$$r^{k+1} = -(\nabla f)(\gamma^l + z^{k+1}) \quad (3.25)$$

$$\text{err} = \frac{\langle r^{k+1}, r^{k+1} \rangle}{\langle r^k, r^k \rangle} \quad (3.26)$$

*if err < tolerance, terminate k loop, else*

$$\beta_k = \frac{1}{\langle r^{k+1}, r^{k+1} \rangle} \quad (3.27)$$

$$p^{k+1} = p^k + \beta_k r^{k+1} \quad (3.28)$$

*continue k loop*

$$\gamma^{l+1} = \gamma^l + z^{k+1} \quad (3.29)$$

*continue l loop*

This algorithm avoids computation of the Hessian but may not always yield a solution. If  $H$  is not positive definite, then the scalar  $\alpha_k$  may be infinite or negative, which violates the conditions under which the algorithm is derived. This may cause an unpredictable breakdown in the one-dimensional minimization algorithm used to find equation (3.23). Thus, although the algorithm presented in equations (3.20)–(3.29) is computationally efficient, the low reliability would require a constant user interface and an additional decision concerning the contingency of breakdown of the algorithm. An alternative algorithm that does not suffer from these disadvantages is presented next.

If the Hessian is nonsingular, but possibly indefinite, then an algorithm that iterates  $K$  times with  $L$  restarts is given as follows:

*begin loop for  $l=1, \dots, L$*

*compute  $G_l = -(\nabla f)(\gamma^l)$ ,  $H_l = (\nabla \nabla f)(\gamma^l)$  and set  $z^1 = 0$*

$$r^1 = G_l \quad (3.30)$$

$$\beta_0 = \frac{1}{\langle H_l r^1, H_l r^1 \rangle} \quad (3.31)$$

$$p^1 = \beta_0 H_l r^1 \quad (3.32)$$

*begin loop for  $k=1, \dots, K$*

$$\alpha_k = \frac{1}{\langle H_l p^k, H_l p^k \rangle} \quad (3.33)$$

$$z^{k+1} = z^k + \alpha_k p^k \quad (3.34)$$

$$r^{k+1} = r^k - \alpha_k H_l p^k \quad (3.35)$$

$$err = \frac{\langle r^{k+1}, r^{k+1} \rangle}{\langle r^k, r^k \rangle} \quad (3.36)$$

*if err < tolerance, terminate k loop, else*

$$\beta_k = \frac{1}{\langle H_l r^{k+1}, H_l r^{k+1} \rangle} \quad (3.37)$$

$$p^{k+1} = p^k + \beta_k H_l r^{k+1} \quad (3.38)$$

*continue k loop*

$$\gamma^{l+1} = \gamma^l + z^{k+1} \quad (3.39)$$

*continue l loop*

This algorithm requires knowledge of both the gradient and Hessian of the function but offers greater reliability against failure.

The quadratic is a local approximation, so that if the initial guess is not near a solution to the global problem, then there is a point of diminishing returns in terms of minimizing the quadratic. In other words, if the quadratic is not near a minimum of the original function, then it is not beneficial to minimize it very far. It is better to perform a small number of conjugate gradient iterations on the quadratic and then restart. As the quadratic approximation gets closer to the actual trough of the original function, it becomes advantageous to increase the number of conjugate gradient iterations. In general, the number of restarts  $L$  will be a large number such that  $L \gg 1$ . This contributes to the majority of the computation, because it takes much more time to compute the gradient and Hessian than to perform a conjugate gradient iteration. The conservative approach would be to always keep  $K$  small such that  $1 < K < 10$ . However, the optimal approach would be to allow  $K$  to be a function of  $L$ , such that  $K = K(L)$  could be incremented when the quadratic gets closer to a solution region.

#### IV. MINIMIZING MAXIMUM SIDELobe LEVEL

Now that the numerical algorithm has been established, it is necessary to find the appropriate function to minimize. The desired goal of the synthesis is to minimize the peak sidelobe power level while keeping the peak of the main beam as high as possible. It is well known that the minimum peak sidelobe level for a given beam width occurs when all the sidelobes are at the same height. Therefore, in order to implement the algorithm of the previous section, it is necessary to find a suitable function that, when minimized, will yield this behavior. The function used by Deford and Gandhi (refs. 4 and 5) is given by

$$f(\gamma) = \frac{P(\gamma, u_i, v_i)}{P(\gamma, 0, 0)} \quad (4.1)$$

where  $P(\gamma, u_i, v_i)$  is the peak sidelobe power level located at the point  $(u_i, v_i)$  and  $P(\gamma, 0, 0)$  is the power at the main beam peak. If this function is used, then as the iterations proceed, a new peak sidelobe is found and the function is changed. Unfortunately, using this approach can lead to stability problems as the sidelobe level reaches a uniform level. This is because as the sidelobe level becomes more uniform, the pattern becomes very sensitive to a change in any sidelobe. This can lead to oscillations and instability.

The approach proposed in this report is to minimize the weighted average of the current peak sidelobe level with all previous peak sidelobe levels at each restart of the algorithm given by equations (3.30) through (3.39). More specifically, at each restart, the sidelobe region is searched to find the peak sidelobe point. If the peak sidelobe point occurs at a previous peak, then the components of the weighted average remain unchanged. However, if the peak occurs at a new sample point, then a new function is formed by taking the weighted average of the new point with all the previous peak points. As an example, suppose that at some restart number, the sidelobe region is searched and the point  $(u_k, v_k)$ ,  $k \neq i$  is found to have the highest sidelobe.

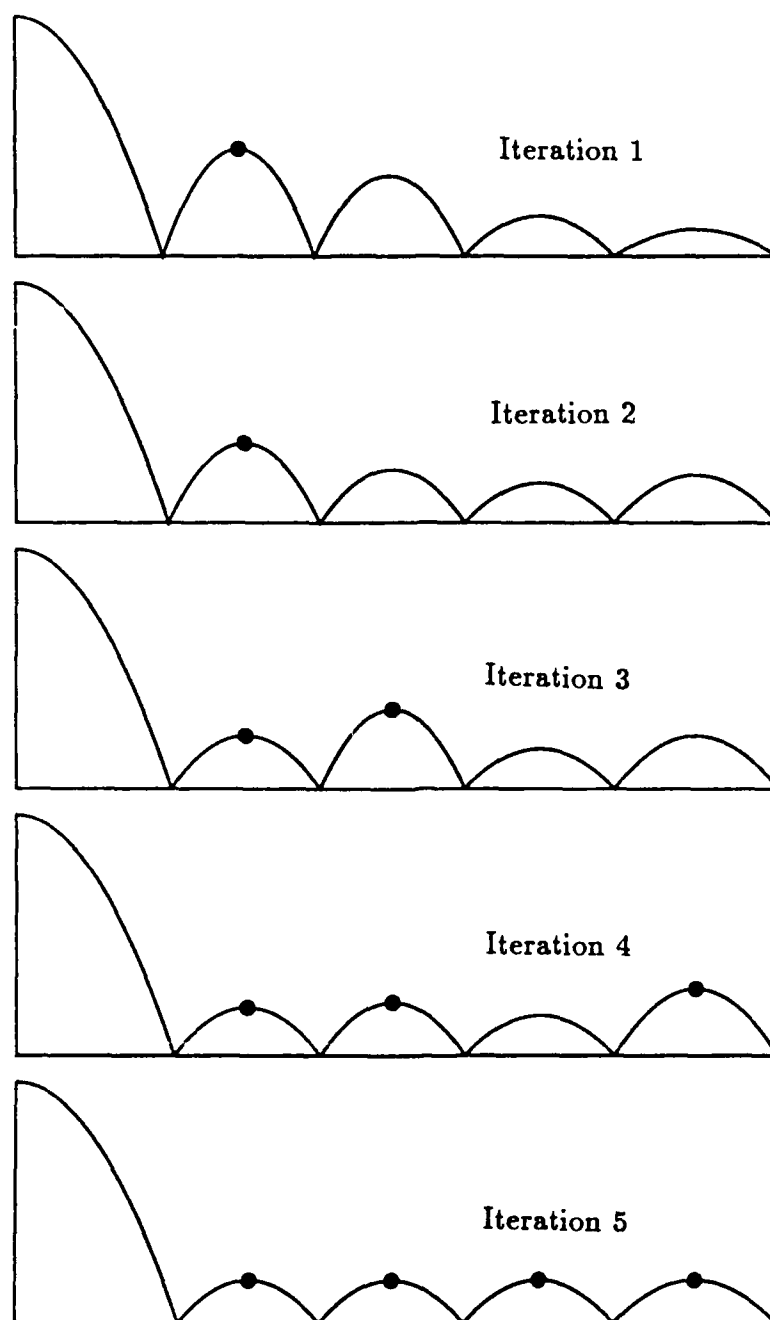


Fig. 2. Weighted Average Minimization

Then equation (4.1) would be modified to

$$f(\gamma) = \frac{1}{P(\gamma, 0, 0)} \frac{\sum_{i=1}^2 P^2(\gamma, u_i, v_i)}{\sum_{i=1}^2 P(\gamma, u_i, v_i)} \quad (4.2)$$

The averaging effect is chosen because as the sidelobes approach a uniform level, the averaging provides an automatic damping effect that reduces the sensitivity of the sidelobe pattern to small changes in the phase. The average is weighted so that more emphasis is placed on minimizing the largest sidelobes.

Figure 2 exhibits the typical evolution of an axial cut in the sidelobe pattern, assuming that the synthesis produced uniform sidelobes in five iterations. At the first iteration, the near-in sidelobe has the highest peak. After the first minimization, the same point still has the highest peak. At the third iteration, the second sidelobe now has the highest level, so a weighted average of the two lobes is formed. At the fourth iteration, the fourth lobe now is the highest, and finally at the fifth iteration, all the lobes are at an equal level. Because of the conservation of energy, this level will always be between the highest and lowest peaks of the pattern with uniform illumination.

A general weighted average function may now be defined as

$$f(\gamma) = \frac{P_a(\gamma)}{P_0(\gamma)} \quad (4.3)$$

where  $P_a(\gamma)$  is the weighted average sidelobe level at the  $l$ th restart

$$P_a(\gamma) = \frac{\sum_{i=1}^{M_l} P_i^2}{\sum_{i=1}^{M_l} P_i} \quad (4.4)$$

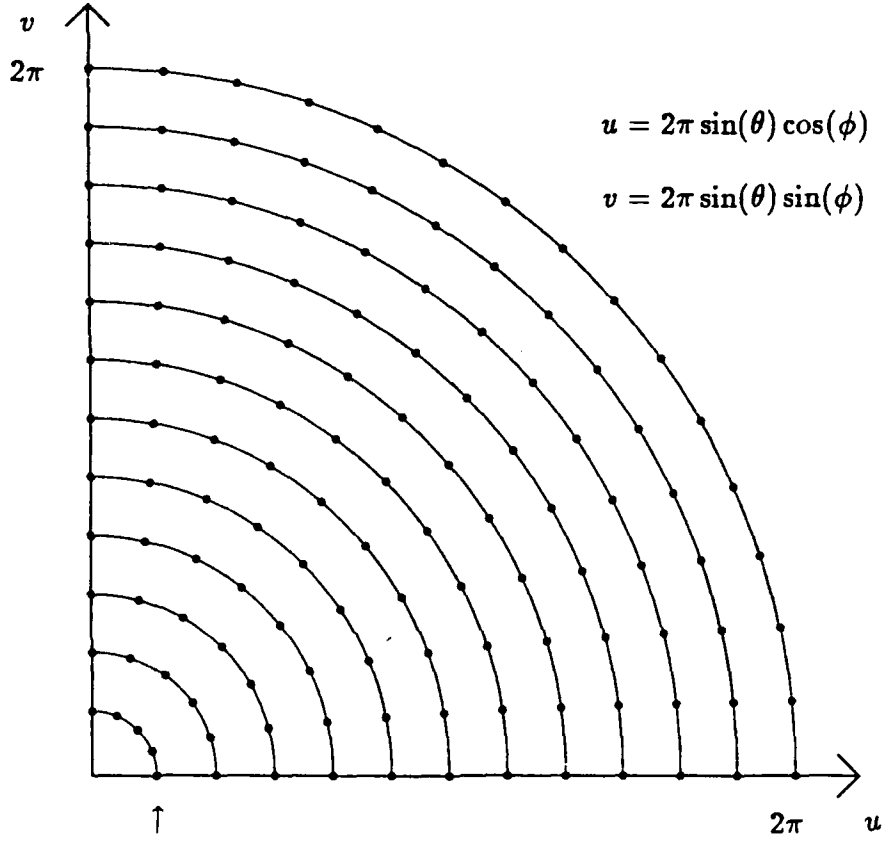
and the  $u-v$  plane is sampled as in Fig. 3. The components of the gradient and Hessian are computed from

$$\frac{\partial f}{\partial \gamma_m} = \frac{1}{P_0} \left[ \frac{\partial P_a}{\partial \gamma_m} - \frac{P_a}{P_0} \frac{\partial P_0}{\partial \gamma_m} \right] \quad (4.5)$$

$$\frac{\partial^2 f}{\partial \gamma_m \partial \gamma_n} = \frac{-1}{P_0} \left[ \frac{\partial f}{\partial \gamma_n} \frac{\partial P_0}{\partial \gamma_m} + \frac{\partial f}{\partial \gamma_m} \frac{\partial P_0}{\partial \gamma_n} - \frac{\partial^2 P_a}{\partial \gamma_m \partial \gamma_n} + \frac{P_a}{P_0} \frac{\partial^2 P_0}{\partial \gamma_m \partial \gamma_n} \right] \quad (4.6)$$

Defining the  $M \times N$  Jacobian matrix  $J$  as

$$J = \begin{bmatrix} \frac{\partial P_1}{\partial \gamma_1} & \frac{\partial P_1}{\partial \gamma_2} & \dots & \frac{\partial P_1}{\partial \gamma_N} \\ \frac{\partial P_2}{\partial \gamma_1} & \frac{\partial P_2}{\partial \gamma_2} & \dots & \frac{\partial P_2}{\partial \gamma_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial P_M}{\partial \gamma_1} & \frac{\partial P_M}{\partial \gamma_2} & \dots & \frac{\partial P_M}{\partial \gamma_N} \end{bmatrix}, \quad (4.7)$$



Main Beam Null

Fig. 3. Sampled  $u - v$  Plane

the first and second derivatives may be expanded as

$$\frac{\partial P_a}{\partial \gamma_m} = \frac{2}{s_1^2} [s_1 B_m - s_2 A_m] \quad (4.8)$$

$$\frac{\partial^2 P_a}{\partial \gamma_m \partial \gamma_n} = \frac{2}{s_1^2} \left[ A_n B_m - \left[ A_m + \frac{\partial P_a}{\partial \gamma_m} \right] B_n + s_1 Q_{mn} - s_2 W_{mn} \right] \quad (4.9)$$

where

$$s_1 = \sum_{i=1}^M P_i \quad (4.10)$$

$$s_2 = \frac{1}{2} \sum_{i=1}^M P_i^2 \quad (4.11)$$

$$A_m = \sum_{i=1}^M J_{im} \quad (4.12)$$

$$B_m = \sum_{i=1}^M P_i J_{im} \quad (4.13)$$

$$Q_{mn} = \sum_{i=1}^M \left[ J_{im} J_{in} + P_i \frac{\partial^2 P_i}{\partial \gamma_m \partial \gamma_n} \right] \quad (4.14)$$

$$W_{mn} = \sum_{i=1}^M \frac{\partial^2 P_i}{\partial \gamma_m \partial \gamma_n} \quad (4.15)$$

and letting  $\psi_{pq} = u_q x_p + v_q y_p$  yields

$$P_i = \beta_i \left[ \left[ \sum_{k=1}^{4N} I_k \cos(\gamma_k + \psi_{ki}) \right]^2 + \left[ \sum_{k=1}^{4N} I_k \sin(\gamma_k + \psi_{ki}) \right]^2 \right] \quad (4.16)$$

$$\frac{\partial P_i}{\partial \gamma_m} = 2\beta_i I_m \sum_{k=1}^{4N} I_k \sin(\gamma_k - \gamma_m + \psi_{ki} - \psi_{mi}) \quad (4.17)$$

$$\frac{\partial^2 P_i}{\partial \gamma_m \partial \gamma_n} = 2\beta_i I_m I_n \cos(\gamma_n - \gamma_m + \psi_{ni} - \psi_{mi}). \quad (4.18)$$

Note that for the broadside array with a main beam at  $\theta = 0$  and  $\phi = 0$

$$P_i = \beta_i \left[ \left[ \sum_{k=1}^N I_k \zeta_{ik} \cos(\gamma_k) \right]^2 + \left[ \sum_{k=1}^N I_k \zeta_{ik} \sin(\gamma_k) \right]^2 \right] \quad (4.19)$$

$$\frac{\partial P_i}{\partial \gamma_m} = 2\beta_i \zeta_{im} I_m \sum_{k=1}^N I_k \zeta_{ik} \sin(\gamma_k - \gamma_m) \quad (4.20)$$

$$\frac{\partial^2 P_i}{\partial \gamma_m \partial \gamma_n} = 2\beta_i \zeta_{im} \zeta_{in} I_m I_n \cos(\gamma_m - \gamma_n) \quad (4.21)$$

where  $\zeta_{pq} = \cos(u_p x_q) \cos(v_p y_q)$ .

The phase may be initialized as the sum of a linear shift and a perturbation term given by

$$\gamma_k = 2\pi(f_{x0} x_k + f_{y0} y_k) + \xi(x_k, y_k) \quad (4.22)$$

where  $f_{x0} = \sin(\theta_0) \cos(\phi_0)$  and  $f_{y0} = \sin(\theta_0) \sin(\phi_0)$ . The term  $\xi(x_k, y_k)$  must be added if all amplitudes are equal because observation of equation (4.20)

indicates that a linear phase is a stationary point of equation (4.3). Deford and Gandhi (ref. 4) have found that although the final phase distribution is strongly dependent on the initial guess, the resulting peak sidelobe level is not.

It is important from a practical standpoint to have a measure of the ability of the array to concentrate energy in the main beam. The parameter of interest is the directivity defined by

$$\text{Directivity} = \frac{4\pi A_a}{\lambda_0^2} \text{ATE} \cdot \text{PTE} \quad (4.23)$$

where  $A_a$  is the aperture area, ATE is the amplitude taper efficiency, and PTE is the phase taper efficiency. The ATE is a measure of how the amplitude variations of the electric field across the aperture of the array affect the concentration of energy into the main beam. Letting  $s$  denote the surface, the ATE is given by

$$\begin{aligned} \text{ATE} &= \frac{(\iint_s |\bar{E}| ds)^2}{A_a \iint_s |\bar{E}|^2 ds} \\ &\approx \frac{\left(\sum_{k=1}^N I_k\right)^2}{N \sum_{k=1}^N I_k^2} \end{aligned} \quad (4.24)$$

The PTE is a measure of how the phase variations of the electric field across the aperture of the array affect the concentration of energy into the main beam. The PTE is given by

$$\begin{aligned} \text{PTE} &= \frac{|\iint_s \bar{E} ds|^2}{(\iint_s |\bar{E}|^2 ds)^2} \\ &\approx \frac{\left[\sum_{k=1}^N I_k \cos(\gamma_k)\right]^2 + \left[\sum_{k=1}^N I_k \sin(\gamma_k)\right]^2}{\left[\sum_{k=1}^N I_k\right]^2} \end{aligned} \quad (4.25)$$

A phase-only array with a uniform amplitude distribution has an ATE = 1. This implies that the maximum power in the main beam is achieved by a uniform amplitude and phase distribution.

A common characteristic of both amplitude and phase synthesis is that as the sidelobe level is lowered, the beam width increases. This increase in beam width may be measured with respect to the beam width of a uniform array. Assuming a circular aperture of diameter  $D$ , if  $D \gg \lambda_0$ , then the half power beam width (HPBW) of a uniform array is given by

$$\text{HPBW} = 58.95^\circ \frac{\lambda_0}{D} \quad (4.26)$$

and the beam width between first nulls (BWFN) is given by

$$\text{BWFN} = 139.76^\circ \frac{\lambda_0}{D}. \quad (4.27)$$

Note that the beam width is primarily determined by the diameter of the array. It can be observed by Fourier transform theory that the HPBW and the BWFN for a uniform amplitude and phase array are the minimum achievable.

## V. RESULTS

All the results shown in this report were computed by the Fortran 77 program **CGPHSA**. A description of the input parameters as well as the source code listing is given in the appendix. All results were computed on an IBM 3090 computer.

Figure 4 shows the peak sidelobe level as a function of the array diameter for a uniform phase array and a phase-synthesized array. The dipole length is given by  $d_l = 0.5\lambda_0$ , and the center-to-center spacing is given by  $s = 0.577\lambda_0$ . The dipoles are placed at a distance of  $h = 0.25\lambda_0$  above a perfectly conducting ground plane. The number of restarts was approximately  $L = 10N$ , where  $N$  is the number of elements in the first quadrant. The number of conjugate gradient iterations was set at  $K = 2$ . The sidelobe level of the uniform phase array approaches the limit of the continuous distribution, which is approximately  $-17.6$  dB. The synthesized phase array sidelobe level indicates a logarithmic dependence on the array diameter. A logarithmic curve fit yields the equation

$$\text{SLL(dB)} = -4.83 \ln(D) - 10.22. \quad (5.1)$$

Extrapolation based on this equation indicates that a sidelobe level of  $-40$  dB would require about a  $470\lambda_0$  diameter aperture.

The relative aperture efficiency is shown in Fig. 5. The curve follows the same logarithmic behavior as the peak sidelobe. At  $D = 20\lambda_0$ , the power in the main beam has dropped to about 48% of what it is for a uniform distribution. This level is similar to that of an equivalent array using only amplitude tapering.

Figures 6 and 7 compare the half power beamwidth (HPBW) and beamwidth between first nulls (BWFN) of the synthesized array to a uniform array. The synthesized curves appear to be just-scaled versions of equations (4.26) and (4.27). The HPBW for the synthesized array appears to be about

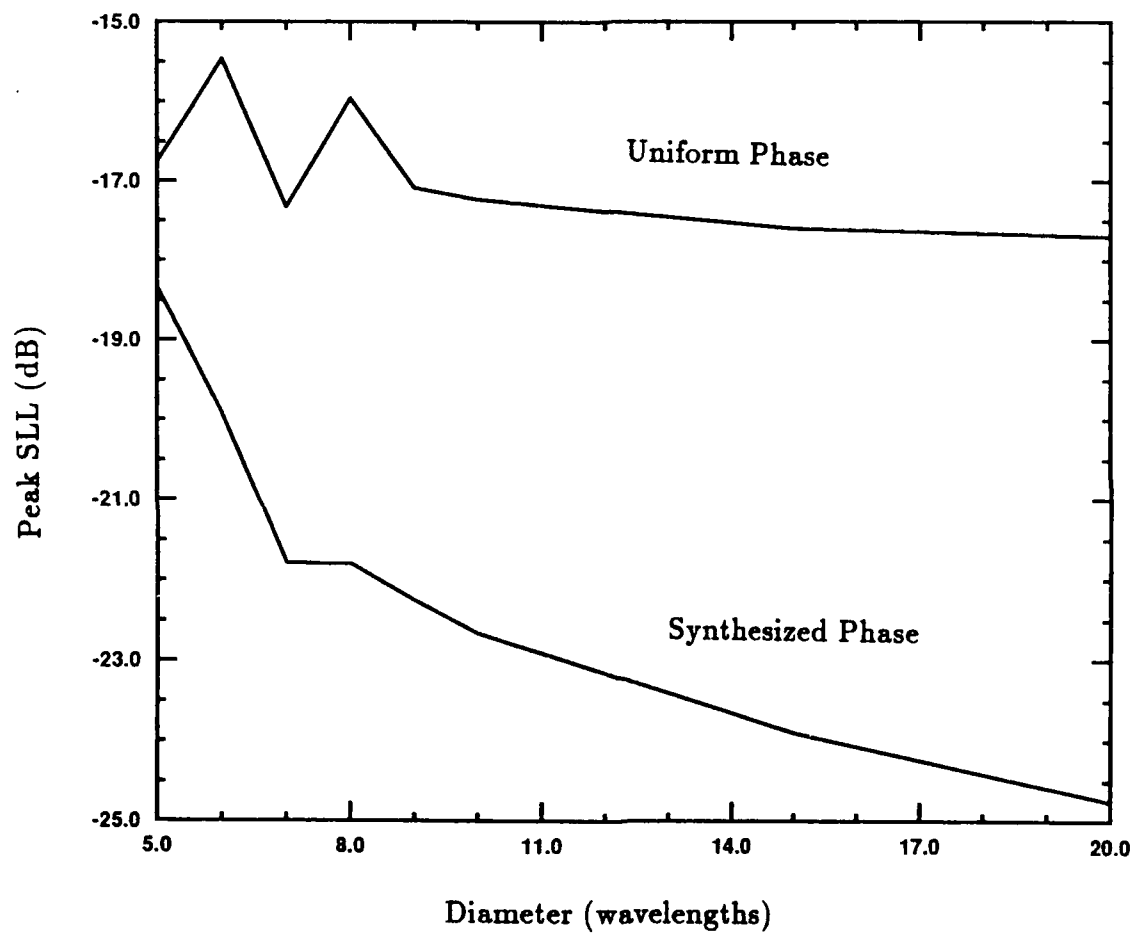


Fig. 4. Peak Sidelobe Level

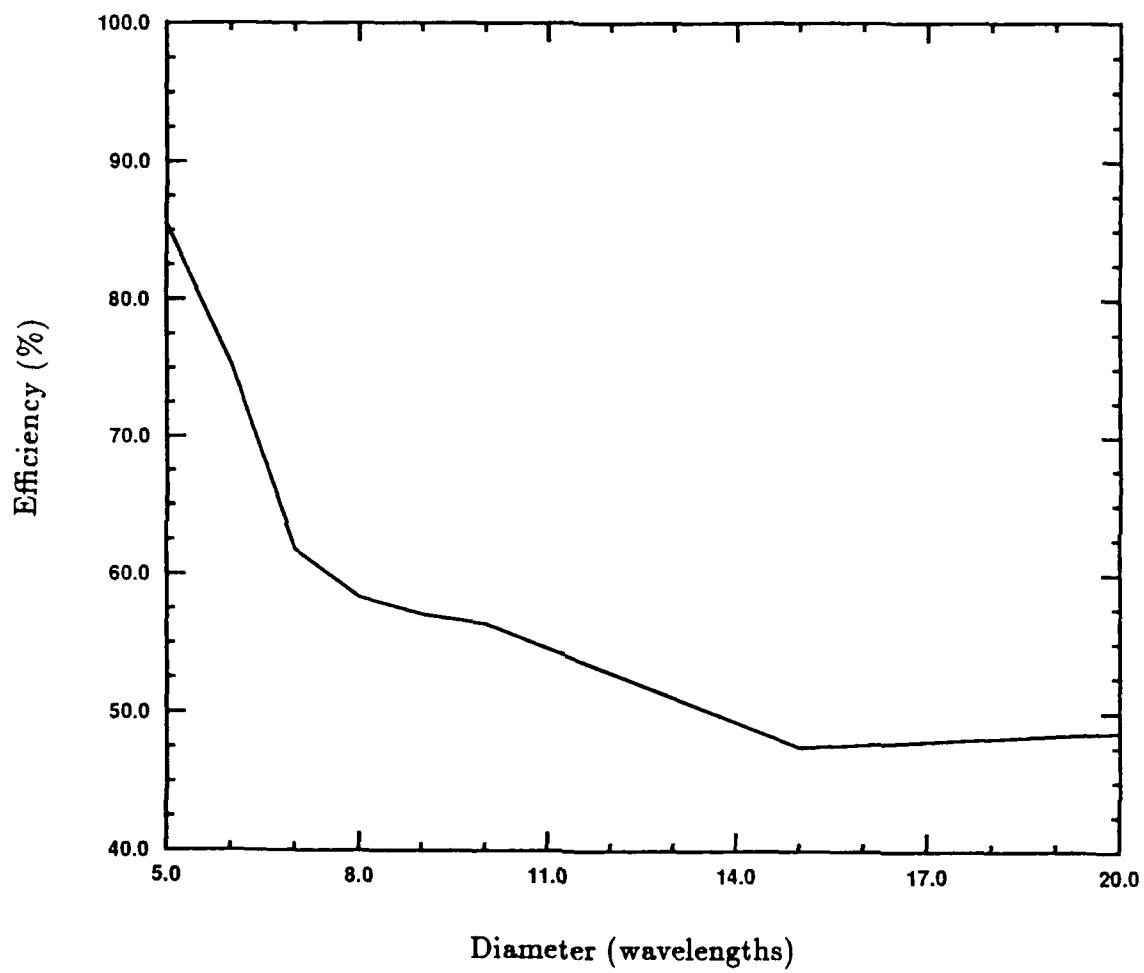


Fig. 5. Aperture Efficiency

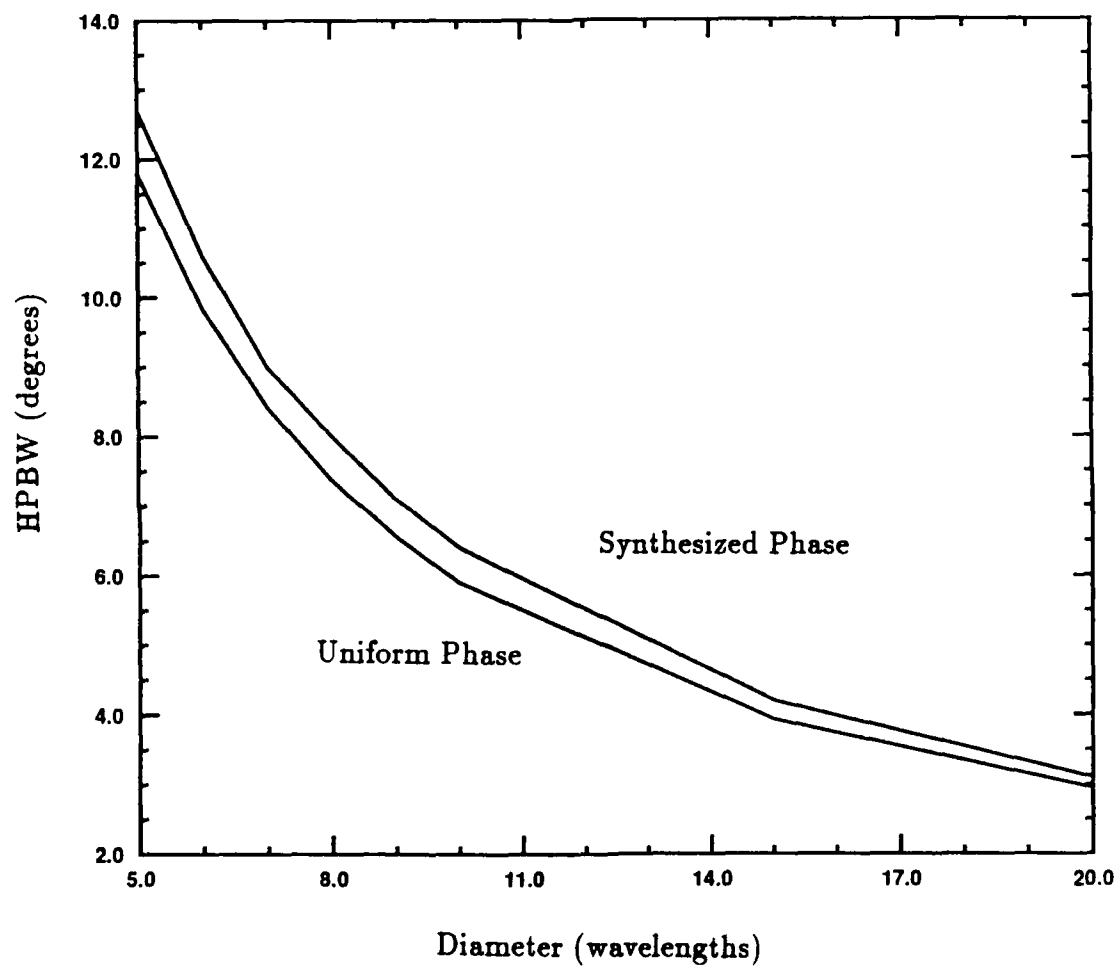


Fig. 6. Half-Power Beam Width

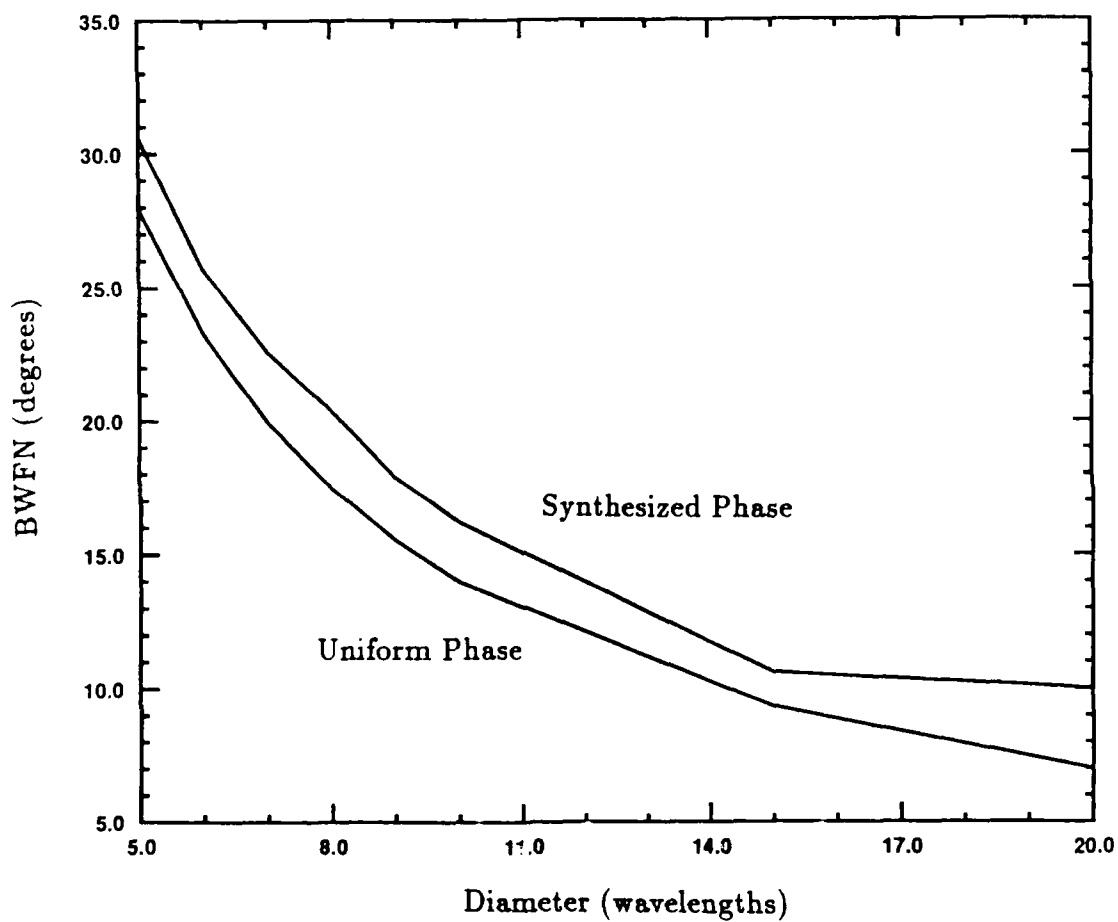


Fig. 7. Beam Width Between First Nulls

1.08 times the uniform HPBW, and the BWFN for the synthesized array is about 1.1 times that of the uniform phase case. This beam broadening is also a characteristic of amplitude tapered arrays and appears to be at a comparable level.

Figures 8 and 9 are power patterns of a  $D = 7\lambda_0$  diameter array plotted in decibels in cylindrical coordinates on a circular base with a -32 dB floor. Figures 10 and 11 are the same patterns plotted in spherical coordinates. As shown in Figs. 8 and 10, the sidelobe level of a uniform phase array decreases monotonically as  $\theta$  goes from  $0^\circ$  to  $90^\circ$ . Figures 9 and 11 indicate that the lowest peak sidelobe level occurs when all the sidelobes have the same height. Therefore, the synthesis of low sidelobes results in a uniform distribution of energy in all sidelobes. The total energy is conserved, because the energy in the peak sidelobes and the main beam is distributed over the lowest sidelobes to bring all the sidelobe peaks to an equal height.

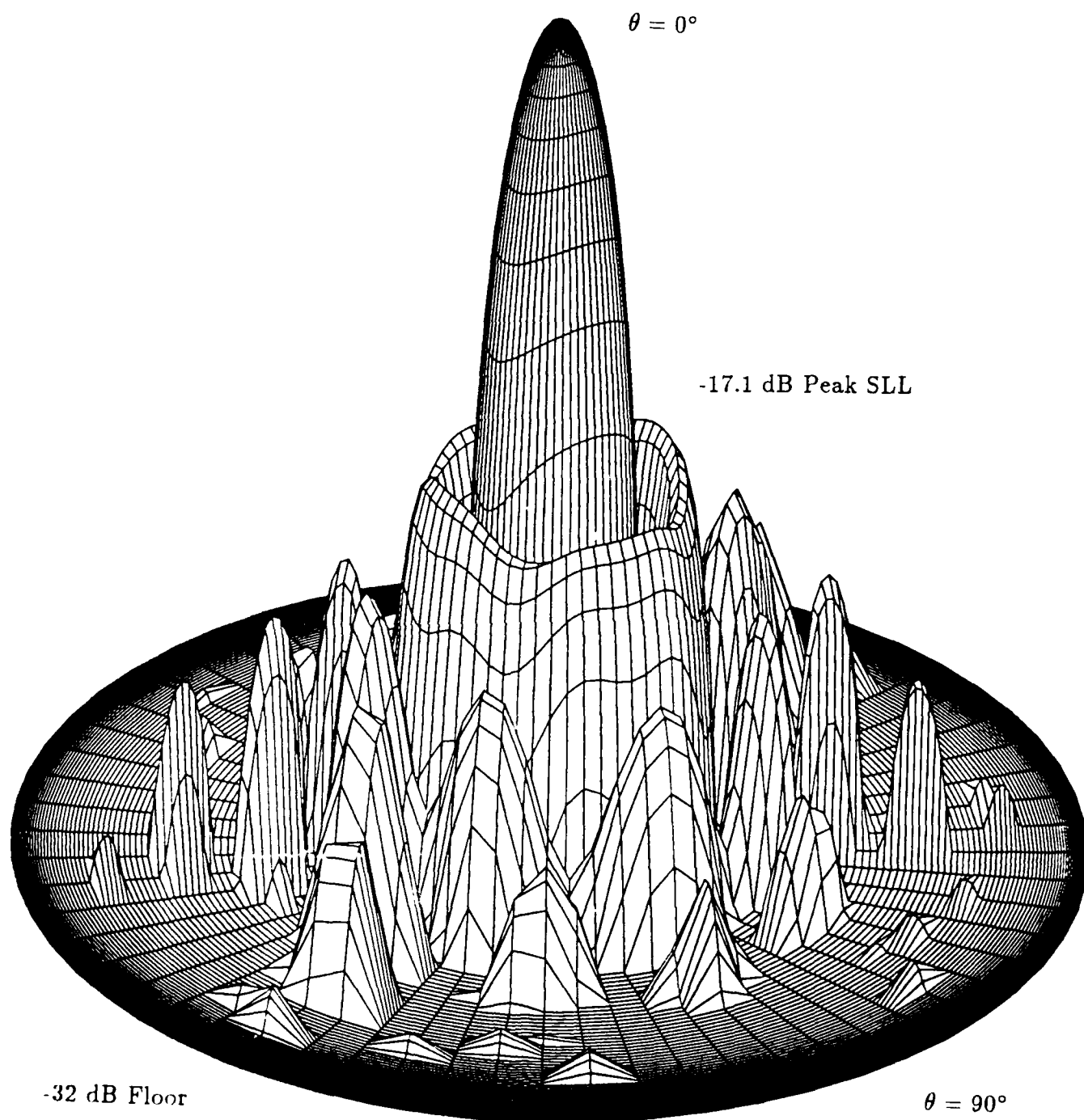


Fig. 8. Uniform Phase Power Pattern in Cylindrical Coordinates

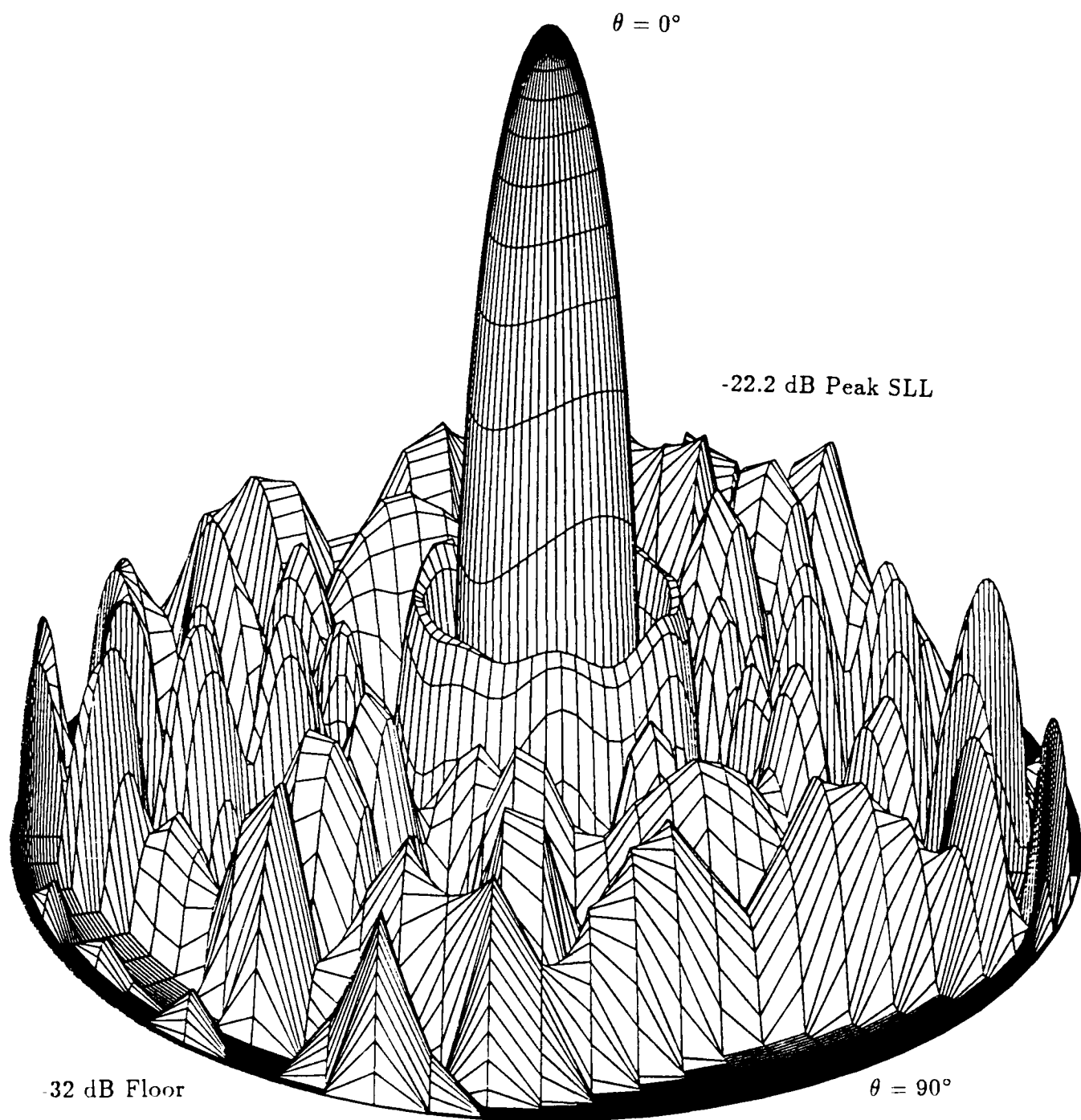


Fig. 9. Synthesized Phase Power Pattern in Cylindrical Coordinates

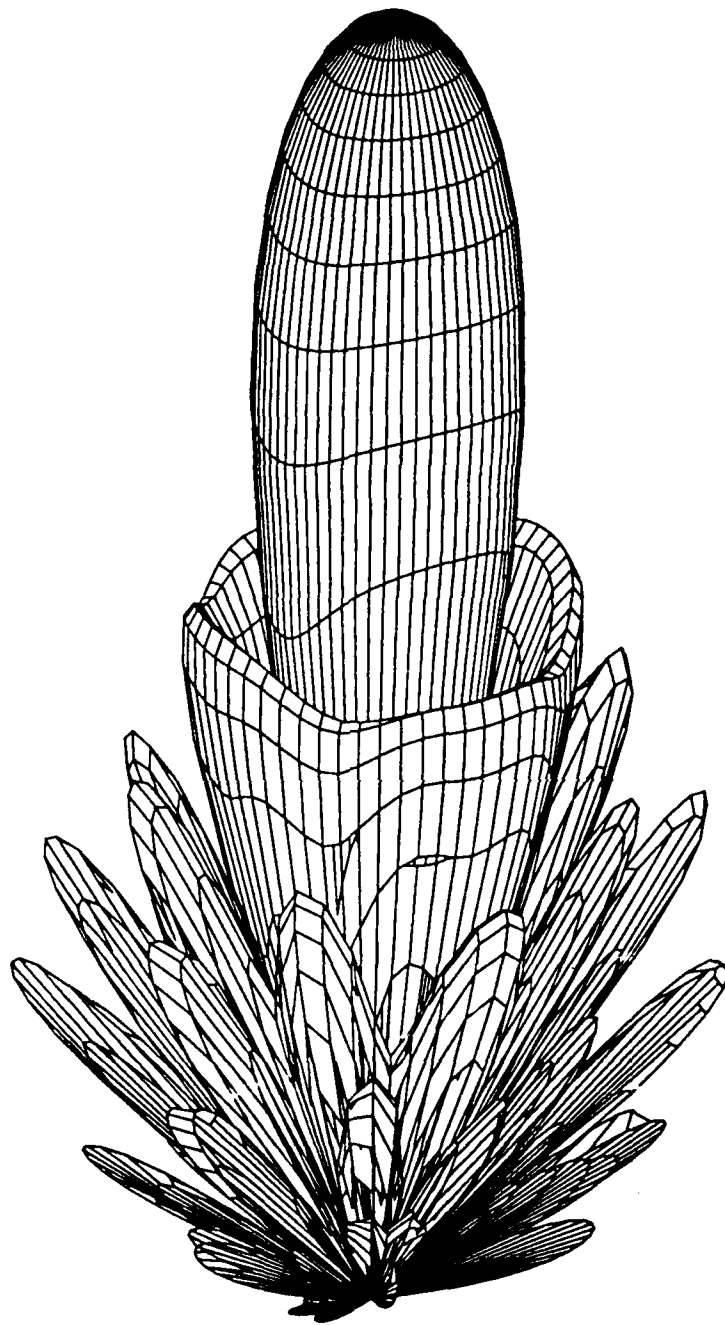


Fig. 10. Uniform Phase Power Pattern in Spherical Coordinates

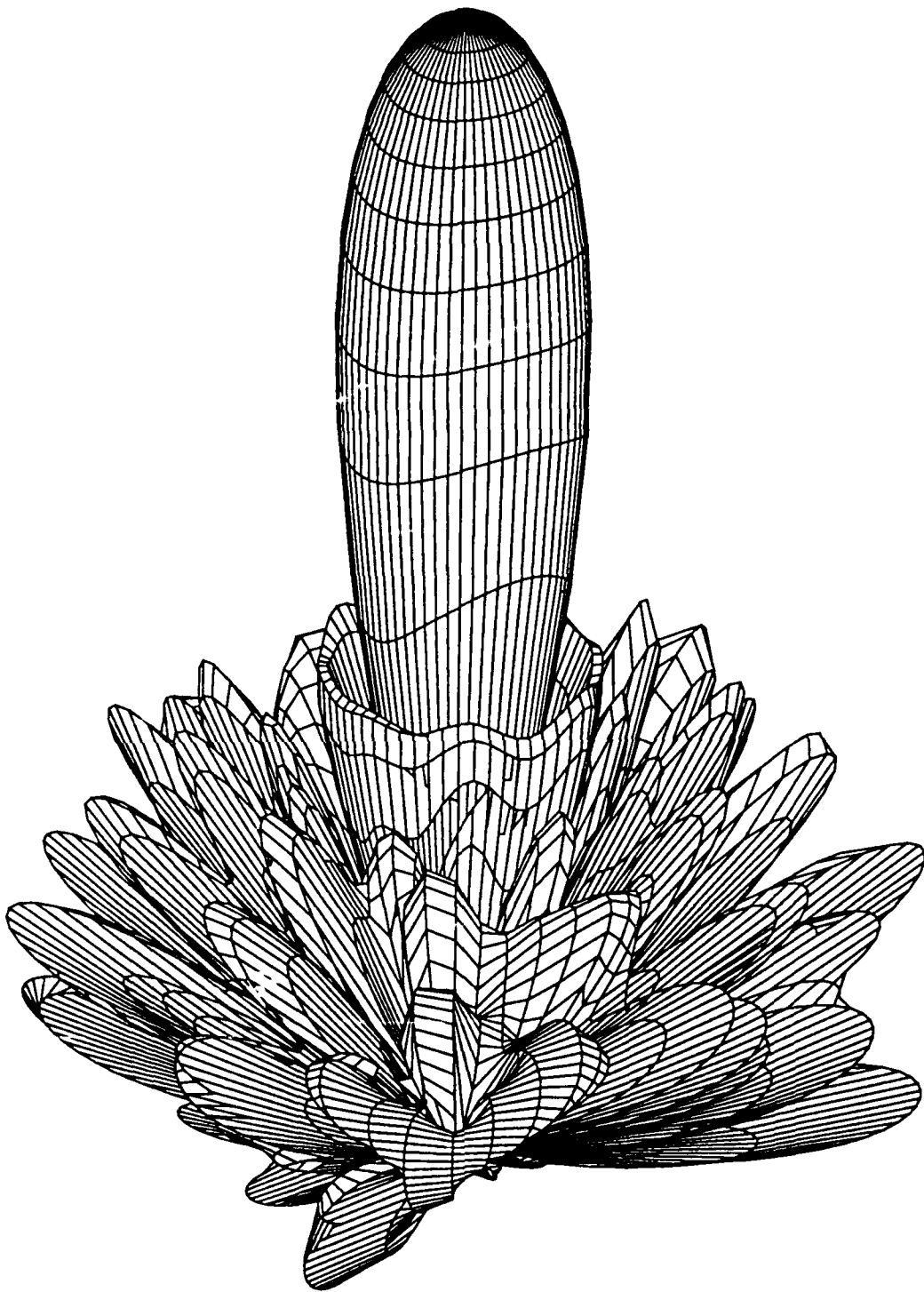


Fig. 11. Synthesized Phase Power Pattern in Spherical Coordinates

## VI. CONCLUSIONS

The application of the conjugate gradient method based algorithm described in this report appears to yield a very reliable numerical procedure. If an accurate initial guess is not available, then the number of restarts must be large and the number of conjugate gradient iterations per restart must be small. The results indicate that the peak sidelobe power level may be decreased by phase-only synthesis. The level appears to be a logarithmic function of the array diameter, which limits the usefulness to very large arrays. The convergence is stabilized by forming a weighted average of present and past peak sidelobe power points.

## REFERENCES

1. M. R. Hestenes and E. Steifel, "Method of Conjugate Gradients for Solving Linear Systems," J. Res. Nat. Bur. Stand., 49(6), 409-436 (Dec. 1952).
2. T. S. Fong and R. A. Birgenheier, "Method of Conjugate Gradients for Antenna Pattern Synthesis," Radio Science, 6(12), 1123-1130 (Dec. 1971).
3. W. H. Press, B. P. Flannery, S. A Teukolsky, and W. T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, MA (Dec. 1986), pp. 301-306.
4. J. F. Deford and O. P. Gandhi, "Phase-Only Synthesis of Minimum Peak Sidelobe Patterns for Linear and Planar Arrays," IEEE Trans. Antennas Propag., 36(2), 191-201 (Feb. 1988).
5. J. F. Deford and O. P. Gandhi, "Mutual Coupling and Sidelobe Tapers in Phase-Only Antenna Synthesis for Linear and Planar Arrays," IEEE Trans. Antennas Propag., 36(11), 1624-1629 (Nov. 1988).

## APPENDIX: FORTRAN 77 PROGRAM

### INPUTS

The Fortran 77 program CGPHSA synthesizes the phase required to yield the lowest sidelobe level for a circular two-dimensional array of x-directed half-wavelength dipoles. Because Fortran 77 allows only static storage allocation, the dimensions of all vectors must appear in the calling program. The required dimension parameters and declarations are given as follows.

```

1 C*****C
2   PROGRAM CGPHSA
3   PARAMETER (M=146,NS=980)
4   REAL*4 X(M),Y(M),A(M),P(M),W(M),R(M),Q(M),Z(M),G(M),E(M),T(M)
5   &,U(NS),V(NS),B(NS),H(M,M),S(M,NS),C(M,M)
6   INTEGER*4 J(NS)
7   D=7.0
8   SX=.577
9   SY=.5
10  CALL GEOMET(D,SX,SY,N,X,Y)
11  NOR=0
12  CALL PARBOL(D,NOR,N,X,Y,A)
13  CALL CGPATT(M,N,X,Y,D,A,NS,P,W,R,Q,Z,G,U,V,T,B,J,E,H,S,C)
14  .
15  .
16  .
17  END
18 C*****C

```

The maximum vector dimension  $M$  is the number of elements that fits into a square aperture of side  $D$ , multiplied by the ratio of the areas of a circle of diameter  $D$  to a square of side  $D$ , and is given by

$$M = \text{nearest integer} \left[ \frac{\pi D^2}{16SX SY} \right] \quad (A.1)$$

The parameter  $NT$  should be set to

$$NT = \text{nearest integer} [6.44D - 5] \quad (A.2)$$

and the parameter  $NS$  is then set to

$$NS = \frac{1}{2} NT(9 + NT) \quad (A.3)$$

The subroutine **GEOMET** returns the number of elements and rectangular coordinates of the center point of all dipoles in the first quadrant of the array. The diameter **r**, center-to-center x-spacing **SX**, and the center-to-center y-spacing **SY** are the required inputs. The following inputs are passed to the subroutine **GEOMET**.

<u>Input</u>	<u>Type</u>	<u>Description</u>
<b>D</b>	<b>R4</b>	diameter of the aperture in wavelengths
<b>SX</b>	<b>R4</b>	center-to-center dipole spacing in the x direction
<b>SY</b>	<b>R4</b>	center-to-center dipole spacing in the y direction
<b>N</b>	<b>I4</b>	unspecified integer
<b>X(M)</b>	<b>R4</b>	unspecified vector
<b>Y(M)</b>	<b>R4</b>	unspecified vector

The following outputs are returned from the subroutine **GEOMET**.

<u>Output</u>	<u>Type</u>	<u>Description</u>
<b>N</b>	<b>I4</b>	number of dipoles in the first quadrant
<b>X(M)</b>	<b>R4</b>	x coordinates of the center of each dipole
<b>Y(M)</b>	<b>R4</b>	y coordinates of the center of each dipole

The subroutine **PARBOL** returns the parabolic amplitude distribution of the elements. The diameter and the order of the parabolic taper **NOR** are the required inputs. The following inputs are passed to the subroutine **PARBOL**.

<u>Input</u>	<u>Type</u>	<u>Description</u>
<b>D</b>	<b>R4</b>	diameter of the aperture in wavelengths
<b>NOR</b>	<b>I4</b>	order of the parabolic taper
<b>N</b>	<b>I4</b>	number of dipoles in the first quadrant
<b>X(M)</b>	<b>R4</b>	x coordinates of the center of each dipole
<b>Y(M)</b>	<b>R4</b>	y coordinates of the center of each dipole
<b>A(M)</b>	<b>R4</b>	unspecified vector

The following output is returned from the subroutine **PARBOL**.

<u>Output</u>	<u>Type</u>	<u>Description</u>
<b>A(M)</b>	<b>R4</b>	current amplitudes of each dipole

The following inputs are passed to the subroutine CGPATT.

<u>Input</u>	<u>Type</u>	<u>Description</u>
D	R4	diameter of the aperture in wavelengths
M	I4	maximum dimension of vectors
N	I4	number of dipoles in the first quadrant
NS	I4	total number of sample points in the u-v plane
X(M)	R4	x coordinates of the center of each dipole
Y(M)	R4	y coordinates of the center of each dipole
A(M)	R4	current amplitudes of each dipole
P(M)	R4	scratch vector
W(M)	R4	scratch vector
R(M)	R4	scratch vector
Q(M)	R4	scratch vector
Z(M)	R4	scratch vector
G(M)	R4	scratch vector
T(M)	R4	scratch vector
U(NS)	R4	scratch vector
V(NS)	R4	scratch vector
J(NS)	R4	scratch vector
B(NS)	R4	scratch vector
E(M)	R4	scratch matrix
H(M,M)	R4	scratch matrix
S(M,NS)	R4	scratch matrix
C(M,M)	R4	scratch matrix

## OUTPUTS

The following outputs are returned from the subroutine CGPATT.

<u>Output</u>	<u>Description</u>	<u>Unit</u>
P(N)	relative phases of each dipole	radians
T(N)	relative phases of each dipole	degrees
Z(1)	half-power beam width	degrees
Z(2)	beam width between first nulls	degrees
Z(3)	amplitude taper efficiency	percent
Z(4)	phase taper efficiency	percent

# LISTING

```

1      PROGRAM CGPHSA
2  C*****C
3  C THIS PROGRAM COMPUTES THE PHASES REQUIRED FOR A CIRCULAR PLANAR      C
4  C ARRAY OF X-DIRECTED HALF WAVELENGTH DIPOLES PLACED A QUARTER OF A  C
5  C WAVELENGTH ABOVE A GROUND PLANE.                                  C
6  C*****C
7  C TIMOTHY J. PETERS                                           LAST UPDATED  C
8  C THE AEROSPACE CORPORATION                                05/12/89      C
9  C ELECTRONICS RESEARCH LABORATORY                             C
10 C ELECTROMAGNETIC SCIENCES DEPARTMENT                         C
11 C 2350 EAST EL SEGUNDO BOULEVARD                             C
12 C EL SEGUNDO, CA 90245                                       C
13 C*****C
14 C VARIABLE DESCRIPTIONS:                                       C
15 C                                                             C
16 C NAME      TYPE      DESCRIPTION                             C
17 C                                                             C
18 C D          R4  ARRAY DIAMETER IN WAVELENGTHS                C
19 C SX         R4  CENTER TO CENTER DIPOLE SPACING IN THE X DIRECTION C
20 C SY         R4  CENTER TO CENTER DIPOLE SPACING IN THE Y DIRECTION C
21 C M          I4  MAXIMUM DIMENSION OF VECTORS COMPUTED FROM THE FORMULA C
22 C              M=NEAREST INTEGER(PI*D*D/(16*SX*SY))          C
23 C N          I4  NUMBER OF UNKNOWN ELEMENTS (N <= M)          C
24 C NT         I4  NUMBER OF RADIAL SAMPLE RINGS COMPUTED FROM THE FORMULA C
25 C              NT=NEAREST INTEGER(6.44*D-5)                  C
26 C NS         I4  NUMBER OF SAMPLE POINTS IN U-V PLANE COMPUTED FROM C
27 C              FORMULA NS=NT(9+NT)/2                          C
28 C X(M)       R4  X COORDINATES OF THE CENTER OF EACH DIPOLE    C
29 C Y(M)       R4  Y COORDINATE OF THE CENTER OF EACH DIPOLE    C
30 C A(M)       R4  AMPLITUDES OF EACH DIPOLE                    C
31 C P(M)       R4  SCRATCH VECTOR                                C
32 C W(M)       R4  SCRATCH VECTOR                                C
33 C R(M)       R4  SCRATCH VECTOR                                C
34 C Q(M)       R4  SCRATCH VECTOR                                C
35 C Z(M)       R4  SCRATCH VECTOR                                C
36 C G(M)       R4  SCRATCH VECTOR                                C
37 C T(M)       R4  SCRATCH VECTOR                                C
38 C U(NS)      R4  SCRATCH VECTOR                                C
39 C V(NS)      R4  SCRATCH VECTOR                                C
40 C J(NS)      I4  SCRATCH VECTOR                                C
41 C B(NS)      R4  SCRATCH VECTOR                                C
42 C E(K)       R4  SCRATCH VECTOR                                C
43 C H(M,M)     R4  SCRATCH MATRIX                                C
44 C S(M,NS)    R4  SCRATCH MATRIX                                C
45 C C(K,M)     R4  SCRATCH MATRIX                                C
46 C*****C

```

```

47     PARAMETER (M=29,NS=980)
48     REAL*4 X(M),Y(M),A(M),P(M),W(M),R(M),Q(M),Z(M),G(M),T(M),U(NS)
49     &,V(NS),B(NS),E(M),H(M,M),S(M,NS),C(M,M)
50     INTEGER*4 J(NS)
51     OPEN(UNIT=2,FILE='CGPOUT',STATUS='UNKNOWN')
52 C*****
53 C COMPUTE THE COORDINATES OF EACH DIPOLE EXCLUDING THE REFERENCE.      C
54 C THE TOTAL NUMBER OF DIPOLES SHOULD ALREADY BE KNOWN AS NSE.          C
55 C*****
56     D=7.0
57     SX=0.577
58     SY=0.577
59     CALL GEOMET(D,SX,SY,N,X,Y)
60 C*****
61 C COMPUTE THE EXCITATION AMPLITUDE OF EACH DIPOLE                      C
62 C*****
63     NOR=0
64     CALL PARBOL(D,NOR,N,X,Y,A)
65 C*****
66 C COMPUTE THE SOLUTION                                                  C
67 C*****
68     CALL CGPATT(M,N,X,Y,D,A,NS,P,W,R,Q,Z,G,U,V,T,B,J,E,H,S,C)
69 C*****
70 C PRINT THE PHASE IN DEGREES.                                          C
71 C*****
72     WRITE(2,103)
73     WRITE(2,104) (I,P(I),I=1,N)
74     WRITE(2,105) Z(1)
75     WRITE(2,106) Z(2)
76     WRITE(2,107) Z(3)
77     WRITE(2,108) Z(4)
78 C*****
79 C FORMATS                                                                C
80 C*****
81 103  FORMAT(1H1,27X,'COMPUTED PHASE OF EACH ELEMENT IN RADIANs',//
82     &,6X,'ELEMENT',6X,'PHASE',7X,'ELEMENT'
83     &,6X,'PHASE',7X,'ELEMENT',6X,'PHASE',7X,'ELEMENT'
84     &,6X,'PHASE',/,6X,'NUMBER',17X
85     &,'NUMBER',17X,'NUMBER',17X
86     &,'NUMBER',/)
87 104  FORMAT(6X,I4,6X,F7.2,2X,I4,6X,F7.2,2X,I4,6X,F7.2,2X,I4,6X,F7.2)
88 105  FORMAT(10X,'HALF POWER BEAMWIDTH (DEGREES) ' 2X,F6.2)
89 106  FORMAT(10X,'BEAMWIDTH BETWEEN FIRST NULLS (DEGREES)' 2X,F6.2)
90 107  FORMAT(10X,'AMPLITUDE TAPER EFFICIENCY (PERCENT) ' 2X,F6.2)
91 108  FORMAT(10X,'PHASE TAPER EFFICIENCY (PERCENT) ' 2X,F6.2)
92     END
93 C
94     SUBROUTINE GEOMET(D,SX,SY,N,X,Y)

```

```

95 C*****C
96 C THIS SUBROUTINE COMPUTES THE COORDINATES OF THE CENTER OF EACH C
97 C X DIRECTED DIPOLE FILLING THE FIRST QUADRANT OF A CIRCULAR APERTURE C
98 C OF DIAMETER D WITH A DIPOLE LENGTH OF DL. THE CENTER TO CENTER C
99 C X SPACING IS SX AND THE CENTER TO CENTER Y SPACING IS SY. C
100 C*****C
101 REAL*4 X(*),Y(*)
102 REAL*8 DX,DY,XMIN,YMIN,XS,YS,D2,DL2,AR,RC
103 DL=0.5
104 XMIN= SX/2.0
105 YMIN= SY/2.0
106 DX= SX
107 DY= SY
108 DL2= DL/2.0
109 D2= D/2.0
110 SM= AMIN1(SX,SY)
111 NS= INT(D/(2.0*SM))+4
112 N=0
113 DO 1 I=0,NS
114 YS=YMIN+I*DY
115 DO 2 J=0,NS
116 XS=XMIN+J*DX
117 AR=(XS+DL2)*(XS+DL2)+YS*YS
118 RC=DSQRT(AR)
119 IF(RC .LE. D2) THEN
120 N=N+1
121 X(N)=XS
122 Y(N)=YS
123 ELSE
124 END IF
125 2 CONTINUE
126 1 CONTINUE
127 RETURN
128 END
129 C
130 SUBROUTINE PARBOL(D,NOR,N,X,Y,A)
131 C*****C
132 C THIS SUBROUTINE COMPUTES THE DIPOLE AMPLITUDE EXCITATION USING A C
133 C PARABOLIC DISTRIBUTION. C
134 C*****C
135 REAL*4 X(*),Y(*),A(*)
136 DO 1 I=1,N
137 R=SQRT(X(I)*X(I)+Y(I)*Y(I))
138 A(I)=(1.0-(2.0*R/D)**2)**NOR
139 1 CONTINUE
140 RETURN
141 END
142 C

```

```

143      SUBROUTINE CGPATT(MDIM,NUE,X,Y,D,AI,NMP,P,W,R,Q,Z,G,U,V,TV,IP,B
144      &,PW,H,PSI,JC)
145 C*****C
146 C THIS SUBROUTINE IMPLEMENTS A CONJUGATE GRADIENT METHOD. C
147 C*****C
148 C VARIABLE DESCRIPTIONS: C
149 C C
150 C NAME TYPE DESCRIPTION C
151 C C
152 C NUE I4 NUMBER OF UNKNOWN ELEMENTS C
153 C X(NUE) R4 X COORDINATES OF THE CENTER OF EACH DIPOLE C
154 C Y(NUE) R4 Y COORDINATE OF THE CENTER OF EACH DIPOLE C
155 C D R4 DIAMETER C
156 C HT R4 HEIGHT OF THE DIPOLE ARRAY ABOVE THE GROUND PLANE J
157 C AI(NUE) R4 AMPLITUDES OF EACH DIPOLE C
158 C NMP I4 NUMBER OF MATCH POINTS IN U-V SPACE C
159 C NJP I4 MAXIMUM NUMBER OF PEAK SIDELobe LEVEL POINTS C
160 C NT I4 NUMBER OF RADIAL SAMPLE RINGS C
161 C IRS I4 NUMBER OF RESTARTS C
162 C P(NUE) R4 SEARCH DIRECTION C
163 C W(NUE) R4 UNKNOWN PHASE AT EACH DIPOLE C
164 C R(NUE) R4 RESIDUAL C
165 C Q(NUE) R4 WORK VECTOR FOR CG ALGORITHM C
166 C Z(NUE) R4 ACTUAL UNKNOWN VECTOR FOR THE CG ALGORITHM C
167 C G(NUE) R4 GRADIENT C
168 C U(NMP) R4 U COORDINATES OF THE SAMPLE SPACE C
169 C V(NMP) R4 V COORDINATES OF THE SAMPLE SPACE C
170 C TV(NUE) R4 TEMPORARY VECTOR THAT STORES THE PHASE AT LOWEST SLL C
171 C IP(NMP) I4 TAG ARRAY CONTAINING ALL PREVIOUS PEAK SLL LOCATIONS C
172 C B(NMP) R4 BETA C
173 C PW(NJP) R4 POWER AT EACH PEAK SLL LOCATION C
174 C H(NUE,NUE) R4 HESSIAN C
175 C PSI(NUE,NMP) R4 MULTIPLIER DUE TO ELEMENT PATTERN C
176 C JC(NJP,NUE) R4 JACOBIAN C
177 C*****C
178 C NMP EQUALS THE SUM FROM K=0 TO K=NT-1 OF (IPM+K*IPD) WHICH EQUALS C
179 C NMP=NT*(2*IPM+(NT-1)*IPD)/2
180 C*****C
181 REAL*4 H(MDIM,MDIM),PSI(MDIM,NMP),X(*),Y(*),P(*),W(*),R(*),Q(*)
182 &,Z(*),G(*),B(*),U(*),V(*),PW(*),JC(MDIM,MDIM),AI(*),TV(*)
183 INTEGER*4 IP(*)
184 HT=0.25
185 NJP=MDIM
186 NT=NINT(6.44*D-5)
187 IRS=10*NUE
188 SLLMIN=1000.0
189 RAD=.17453293E-01
190 PI=.3141593E+01

```

```

191      TP=.62831853E+01
192 C*****C
193 C COMPUTE THE AMPLITUDE TAPER LOSS. C
194 C*****C
195      CALL AMPTAP(NUE,AI,AL)
196 C*****C
197 C INITIALIZE THE ELEMENT PHASES USING A LINEAR PHASE AND PERTURBATION. C
198 C*****C
199      CALL INITAL(NUE,D,X,Y,W)
200 C*****C
201 C FIND THE BEAMWIDTH OF THE INITIAL PATTERN. C
202 C*****C
203      CALL BEAMWD(NUE,X,Y,AI,W,HT,HPBW,BWFN)
204      DO 55 IJ=1,NUE
205          Z(IJ)=0.0
206 55  CONTINUE
207      CALL BEAMWD(NUE,X,Y,AI,Z,HT,HPBWZ,BWFNZ)
208 C*****C
209 C COMPUTE THE SAMPLE POINTS IN U-V SPACE. C
210 C*****C
211      TMIN=BWFN/2.0
212      TMAX=89.0
213      PMIN=0.0
214      PMAX=90.0
215      IPM=5
216      IPD=1
217      CALL ANGLES(TMIN,TMAX,PMIN,PMAX,N1,IPM,IPD,U,V)
218      CALL BETASO(HT,BO)
219      CALL BETASI(NMP,U,V,HT,B)
220      CALL PSIMNI(MDIM,NUE,NMP,X,Y,U,V,PSI)
221      CALL POWERO(NUE,Z,AI,BO,P01)
222      INIT=0
223      CALL POWMAX(MDIM,NUE,NMP,INIT,Z,PSI,B,AI,IP,NP,LMAX1,PMAX1)
224      P1DB=10.0*ALOG10(PMAX1/P01)
225      INIT=0
226      CALL POWMAX(MDIM,NUE,NMP,INIT,W,PSI,B,AI,IP,NP,LMAX1,PMAX1)
227      NP=1
228      DO 1 IN=1,IRS
229          CALL POWERO(NUE,W,AI,BO,P0)
230          IF(IN.EQ. 1) THEN
231              INIT=0
232          ELSE
233              INIT=1
234          END IF
235          IF(NP.EQ. NJP) THEN
236              GO TO 909
237          ELSE
238              END IF

```

```

239      CALL POWMAX(MDIM,NUE,NMP,INIT,W,PSI,B,AI,IP,NP,L,PMAX)
240      CALL POWPIE(MDIM,NUE,NMP,W,PSI,B,AI,NP,IP,S1,S2,PA,PW)
241      CALL JACOBI(MDIM,NUE,NMP,NJP,W,PSI,B,IP,NP,S1,S2,PW,R,Z,P,AI,JC)
242      CALL PPORGM(NUE,W,X,Y,AI,BO,Q)
243      CALL GRADFG(NUE,PO,PA,P,Q,G)
244      CALL HESIAN(MDIM,NUE,NMP,NJP,W,X,Y,PSI,BO,B,IP,NP,PO,PA,S1,S2
245      &,R,Z,P,Q,PW,JC,G,AI,H)
246 C*****C
247 C INITIALIZE THE RESIDUAL C
248 C*****C
249      DO 2 I=1,NUE
250          R(I)=-G(I)
251      2 CONTINUE
252      IF(IN.EQ. 1) THEN
253          CALL NORM22(NUE,R,R0)
254      ELSE
255          END IF
256      DO 3 I=1,NUE
257          Z(I)=0.0
258      3 CONTINUE
259 C*****C
260 C INITIALIZE SEARCH VECTOR C
261 C*****C
262      CALL MATVEC(MDIM,NUE,H,R,Q)
263      CALL NORM22(NUE,Q,QN)
264      BEO=1.0/QN
265      DO 4 I=1,NUE
266          P(I)=BEO*Q(I)
267      4 CONTINUE
268 C*****C
269 C PERFORM 2 CONJUGATE GRADIENT ITERATIONS. C
270 C*****C
271      DO 5 K=1,2
272 C*****C
273 C UPDATE PHASE VECTOR AND RESIDUAL C
274 C*****C
275      CALL MATVEC(MDIM,NUE,H,P,Q)
276      CALL NORM22(NUE,Q,QN)
277      AK=1.0/QN
278      DO 6 I=1,NUE
279          Z(I)=Z(I)+AK*P(I)
280      6 CONTINUE
281      DO 7 I=1,NUE
282          R(I)=R(I)-AK*Q(I)
283      7 CONTINUE
284 C*****C
285 C UPDATE SEARCH VECTOR C
286 C*****C

```

```

287      CALL MATVEC(MDIM,NUE,H,R,Q)
288      CALL NORM22(NUE,Q,QN)
289      BEK=1.0/QN
290      DO 8 I=1,NUE
291          P(I)=P(I)+BEK*Q(I)
292      8   CONTINUE
293      5   CONTINUE
294  C*****C
295  C KEEP THE PHASE BETWEEN -PI AND PI.                      C
296  C*****C
297      PI=.31415926E+01
298      TP=.62831853E+01
299      DO 9 I=1,NUE
300          W(I)=W(I)+Z(I)
301  888    CONTINUE
302          IF(W(I) .GT. PI) THEN
303              W(I)=W(I)-TP
304              GO TO 888
305          ELSE
306              END IF
307          IF(W(I) .LT. -PI) THEN
308              W(I)=W(I)+TP
309              GO TO 888
310          ELSE
311              END IF
312      9   CONTINUE
313  C*****C
314  C COMPUTE THE SIDELobe LEVEL AND COMPARE WITH THE MINIMUM ACHIEVED      C
315  C IN ALL PRIOR ITERATIONS.  STORE THE MINIMUM IN A TEMPORARY VECTOR.    C
316  C*****C
317      PDB=10.0*ALOG10(PMAX/PO)
318      IF(PDB .LT. SLLMIN) THEN
319          ISLL=IN
320          SLLMIN=PDB
321      DO 77 I=1,NUE
322          TV(I)=W(I)
323  77    CONTINUE
324      ELSE
325          END IF
326      CALL BEAMWD(NUE,X,Y,AI,W,HT,HPBW,BWFN)
327      PTL=100.0*(PO/PO1)
328      1   CONTINUE
329  909    CONTINUE
330  C*****C
331  C CHANGE THE COMPUTED PHASE FROM RADIANS TO DEGREES                C
332  C*****C
333      CALL DEGREE(NUE,TV,P)
334      Z(1)=HPBW

```

```

335      Z(2)=BWFN
336      Z(3)=ATL
337      Z(4)=PTL
338      RETURN
339      END
340 C
341      SUBROUTINE AMPTAP(NUE,AI,ATL)
342 C*****
343 C THIS SUBROUTINE COMPUTES THE AMPLITUDE TAPER LOSS. C
344 C*****
345      REAL*4 AI(*)
346      SUM1=0.0
347      SUM2=0.0
348      DO 1 I=1,NUE
349          SUM1=SUM1+AI(I)
350          SUM2=SUM2+AI(I)*AI(I)
351 1 CONTINUE
352      ATL=100.0*SUM1*SUM1/(NUE*SUM2)
353      RETURN
354      END
355 C
356      SUBROUTINE ANGLES(TMIN,TMAX,PMIN,PMAX,NT,IPM,IPD,U,V)
357 C*****
358 C NT (I4) - THE NUMBER OF SAMPLES IN THE THETA DIRECTION. C
359 C IPM (I4) - THE MINIMUM NUMBER OF SAMPLES IN THE PHI DIRECTION. C
360 C IPD (I4) - THE INCREMENT IN THE PHI SAMPLES FOR EACH THETA VALUE. C
361 C*****
362 C THE TOTAL NUMBER OF SAMPLE POINTS IS GIVEN BY THE ARITHMETIC C
363 C PROGRESSION FORMULA C
364 C C
365 C SUM FROM K=0 TO K=NT OF (IPM+K*IPD) EQUALS NT*(2*IPM+(NT-1)*IPD)/2 C
366 C*****
367      REAL*4 U(*),V(*)
368      RAD=.17453293E-01
369      TP=.6283185E+01
370      DT=(TMAX-TMIN)/(NT-1)
371      DR=(SIN(RAD*TMAX)-SIN(RAD*TMIN))/NT
372      K=0
373      KI=-IPD
374      DO 1 I=0,NT-1
375          KI=KI+IPD
376          T=TMIN+I*DT
377          RT=RAD*T
378          DP=(PMAX-PMIN)/(KI+IPM-1)
379          DO 2 J=0,KI+IPM-1
380              K=K+1
381              P=PMIN+J*DP
382              RP=RAD*P

```

```

383      ST=SIN(RT)
384      CP=COS(RP)
385      SP=SIN(RP)
386      TPS=TP*ST
387      U(K)=TPS*CP
388      V(K)=TPS*SP
389  2    CONTINUE
390  1    CONTINUE
391      RETURN
392      END
393  C
394      SUBROUTINE INITAL(NUE,D,I,Y,W)
395  C*****C
396  C THIS SUBROUTINE INITIALIZES THE UNKNOWN. C
397  C*****C
398      REAL*4 X(*),Y(*),W(*)
399      TP=.62831853E+01
400      RAD=.17453293E-01
401      RO=0.5*D
402      DO 1 I=1,NUE
403          IS=(-1)**I
404          R=SQRT(X(I)*X(I)+Y(I)*Y(I))
405          W(I)=3.0*RAD*IS
406  1    CONTINUE
407      RETURN
408      END
409  C
410      SUBROUTINE BETASO(HT,B0)
411  C*****C
412  C THIS SUBROUTINE COMPUTES BETA AT THE MAIN BEAM POINT (U0,V0) C
413  C*****C
414      TP2=.39478418E+02
415      ARG1=HT*SQRT(TP2)
416      S=SIN(ARG1)
417      B0=S*S/TP2
418      RETURN
419      END
420  C
421      SUBROUTINE BETASI(NMP,U,V,HT,B)
422  C*****C
423  C THIS SUBROUTINE COMPUTES BETA AT THE ITH SAMPLE POINT (U(I),V(I)) C
424  C*****C
425      REAL*4 U(*),V(*),B(*)
426      TP2=.39478418E+02
427      DO 1 I=1,NMP
428          U2=U(I)*U(I)
429          V2=V(I)*V(I)
430          ARG1=HT*SQRT(TP2-U2-V2)

```

```

431      ARG2=0.25*U(I)
432      S=SIN(ARG1)
433      C=COS(ARG2)
434      B(I)=S*S*C*C/(TP2-U2)
435  1    CONTINUE
436      RETURN
437      END
438  C
439      SUBROUTINE PSIMNI(MDIM,NUE,NMP,X,Y,U,V,PSI)
440  C*****C
441  C THIS SUBROUTINE COMPUTES PSI_{PQ}. C
442  C*****C
443      REAL*4 PSI(MDIM,NMP),X(*),Y(*),U(*),V(*)
444      DO 1 I=1,NUE
445          DO 2 J=1,NMP
446              PSI(I,J)=COS(X(I)*U(J))*COS(Y(I)*V(J))
447  2    CONTINUE
448  1    CONTINUE
449      RETURN
450      END
451  C
452      SUBROUTINE POWERO(NUE,W,AI,B0,PO)
453  C*****C
454  C THIS SUBROUTINE COMPUTES THE MAIN BEAM POWER AT THE POINT (0,0). C
455  C*****C
456      REAL*4 W(*),AI(*)
457      SUM1=0.0
458      SUM2=0.0
459      DO 1 I=1,NUE
460          SUM1=SUM1+AI(I)*COS(W(I))
461          SUM2=SUM2+AI(I)*SIN(W(I))
462  1    CONTINUE
463      PO=B0*(SUM1*SUM1+SUM2*SUM2)
464      RETURN
465      END
466  C
467      SUBROUTINE POWMAX(MDIM,NUE,NMP,INIT,W,PSI,B,AI,IP,NP,LMAX,PMAX)
468  C*****C
469  C THIS SUBROUTINE COMPUTES THE MAXIMUM POWER AND UPDATES THE POINTER C
470  C ARRAY IP WITH ALL THE PREVIOUS MAXIMUM LOCATIONS. C
471  C*****C
472      REAL*4 W(*),PSI(MDIM,NMP),B(*),AI(*)
473      INTEGER*4 IP(*)
474  C*****C
475  C SORT THROUGH THE DATA AND FIND THE MAXIMUM POWER AND THE INDEX C
476  C NUMBER LMAX. C
477  C*****C
478      PMAX=-1000.0

```

```

479      LMAX=-1
480      DO 1 I=1,NMP
481          SUM1=0.0
482          SUM2=0.0
483          DO 2 K=1,NUE
484              SUM1=SUM1+AI(K)*PSI(K,I)*COS(W(K))
485              SUM2=SUM2+AI(K)*PSI(K,I)*SIN(W(K))
486      2      CONTINUE
487      P=B(I)*(SUM1*SUM1+SUM2*SUM2)
488      IF(P .GT. PMAX) THEN
489          LMAX=I
490          PMAX=P
491      ELSE
492          END IF
493      1      CONTINUE
494 C*****C
495 C IF THIS IS THE FIRST CALL THEN SET THE POINTER TO THE FIRST MAXIMUM C
496 C AND RETURN. C
497 C*****C
498      IF(INIT .EQ. 0) THEN
499          IP(1)=LMAX
500      ELSE
501 C*****C
502 C COMPARE LMAX TO ALL THE PREVIOUS POINTERS IP(I),I=1..NP. IF LMAX C
503 C MATCHES ONE OF THE PREVIOUS POINTERS THEN THE POINTER ARRAY NEED C
504 C NOT BE UPDATED. IF LMAX DOES NOT MATCH A PREVIOUS POINTER VALUE C
505 C THEN IT BECOMES THE NEXT VALUE OF THE NP+1TH ELEMENT OF THE POINTER C
506 C ARRAY. C
507 C*****C
508      IS=0
509      DO 3 I=1,NP
510          IF(IP(I) .EQ. LMAX) THEN
511              IS=1
512              GO TO 4
513          ELSE
514              END IF
515      3      CONTINUE
516      4      CONTINUE
517      IF(IS .EQ. 0) THEN
518          NP=NP+1
519          IP(NP)=LMAX
520      ELSE
521          END IF
522      END IF
523      RETURN
524      END
525 C
526      SUBROUTINE POWPIE(MDIM,NUE,NMP,W,PSI,B,AI,NP,IP,S1,S2,PA,PW)

```

```

527 C*****C
528 C THIS SUBROUTINE COMPUTES THE POWER AT THE ITH POINT. C
529 C*****C
530     REAL*4 W(*),PSI(MDIM,NMP),B(*),PW(*),AI(*)
531     INTEGER*4 IP(*)
532     S1=0.0
533     S2=0.0
534     DO 1 I=1,NP
535         INX=IP(I)
536         SUM1=0.0
537         SUM2=0.0
538         DO 2 K=1,NUE
539             SUM1=SUM1+AI(K)*PSI(K,INX)*COS(W(K))
540             SUM2=SUM2+AI(K)*PSI(K,INX)*SIN(W(K))
541     2    CONTINUE
542     PW(I)=B(INX)*(SUM1*SUM1+SUM2*SUM2)
543     S1=S1+PW(I)
544     S2=S2+PW(I)*PW(I)
545     1    CONTINUE
546     PA=S2/S1
547     S2=0.6*S2
548     RETURN
549     END
550 C
551     SUBROUTINE JACOBI(MDIM,NUE,NMP,NJP,W,PSI,B,IP,NP,S1,S2,PW,A,BE
552     &,PAG,AI,JC)
553 C*****C
554 C THIS SUBROUTINE COMPUTES THE FIRST DERIVATIVE OF PI WITH RESPECT C
555 C TO GAMMA_{M}. C
556 C*****C
557     REAL*4 W(*),PSI(MDIM,NMP),B(*),JC(NJP,MDIM),PW(*),A(*),BE(*)
558     &,PAG(*),AI(*)
559     INTEGER*4 IP(*)
560     C1=2.0/S1
561     C2=-2.0*S2/(S1*S1)
562 C*****C
563 C COMPUTE THE (NP X NUE) JACOBIAN C
564 C*****C
565     DO 1 I=1,NP
566         INX=IP(I)
567         TB=2.0*B(INX)
568         DO 2 M=1,NUE
569             SUM1=0.0
570             DO 3 K=1,NUE
571                 SUM1=SUM1+AI(K)*PSI(K,INX)*SIN(W(K)-W(M))
572     3    CONTINUE
573             JC(I,M)=TB*AI(M)*PSI(M,INX)*SUM1
574     2    CONTINUE

```

```

575 1 CONTINUE
576 C*****C
577 C SUM UP THE COLUMNS OF THE JACOBIAN TO USE FOR WORK VECTORS. C
578 C*****C
579 DO 4 M=1,NUE
580 SUM1=0.0
581 SUM2=0.0
582 DO 5 I=1,NP
583 SUM1=SUM1+JC(I,M)
584 SUM2=SUM2+PW(I)*JC(I,M)
585 5 CONTINUE
586 A(M)=SUM1
587 BE(M)=SUM2
588 PAG(M)=C1*SUM2+C2*SUM1
589 4 CONTINUE
590 RETURN
591 END
592 C
593 SUBROUTINE GRADFG(NUE,PO,PA,PAG,POG,G)
594 C*****C
595 C THIS SUBROUTINE COMPUTES THE GRADIENT OF THE WEIGHTED AVERAGE C
596 C MAXIMUM SIDELobe LEVEL POWER. C
597 C*****C
598 REAL*4 PAG(*),POG(*),G(*)
599 C1=1.0/PO
600 C2=-PA/(PO*PO)
601 DO 1 M=1,NUE
602 G(M)=C1*PAG(M)+C2*POG(M)
603 1 CONTINUE
604 RETURN
605 END
606 C
607 SUBROUTINE PPORGM(NUE,W,X,Y,AI,B0,POG)
608 C*****C
609 C THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES OF PO WITH C
610 C RESPECT TO GAMMA_{M} C
611 C*****C
612 REAL*4 W(*),X(*),Y(*),AI(*),POG(*)
613 C=2.0*B0
614 DO 1 M=1,NUE
615 SUM=0.0
616 DO 2 K=1,NUE
617 SUM=SUM+AI(K)*SIN(W(K)-W(M))
618 2 CONTINUE
619 POG(M)=C*AI(M)*SUM
620 1 CONTINUE
621 RETURN
622 END

```

```

023 C
024     SUBROUTINE HESIAN(MDIM,NUE,NMP,NJP,W,X,Y,PSI,B0,B,IP,NP,PO,PA,S1
025     &,S2,A,BE,PAG,POG,PW,JC,G,AI,H)
026 C*****C
027 C THIS SUBROUTINE COMPUTES THE HESSIAN OF THE WEIGHTED AVERAGE      C
028 C MAXIMUM SIDELobe LEVEL POWER.                                     C
029 C*****C
030     REAL*4 H(MDIM,MDIM),W(*),X(*),Y(*),PSI(MDIM,NMP),B(*)
031     &,A(*),BE(*),PAG(*),POG(*),PW(*),JC(NJP,MDIM),G(*),AI(*)
032     INTEGER*4 IP(*)
033     C1=-2.0*PO/(S1*S1)
034     C2=-1.0/(PO*PO)
035     C3=2.0*B0
036 C*****C
037 C COMPUTE ALL INFORMATION FOR THE MTH ROW.                          C
038 C*****C
039     DO 2 M=1,NUE
040         CM=A(M)+PAG(M)
041 C*****C
042 C COMPUTE ALL INFORMATION FOR THE NTH COLUMN.                        C
043 C*****C
044     DO 3 N=M,NUE
045         DWNM=W(N)-W(M)
046         AIMN=AI(M)*AI(N)
047         CDWNM=COS(DWNM)
048         POMN=C3*AIMN*CDWNM
049         P2=PA*POMN
050         SUM1=0.0
051         SUM2=0.0
052         DO 4 I=1,NP
053             INI=IP(I)
054             PIMN=2.0*B(INI)*PSI(N,INI)*PSI(M,INI)*CDWNM
055             SUM1=SUM1+JC(I,M)*JC(I,N)+PW(I)*PIMN
056             SUM2=SUM2+PIMN
057     4     CONTINUE
058         P1=PO*(G(N)*POG(M)+G(M)*POG(N))
059         P3=C1*(A(N)*BE(M)-CM*BE(N)+S1*SUM1-S2*AIMN*SUM2)
060         H(M,N)=C2*(P1+P2+P3)
061     3     CONTINUE
062     2     CONTINUE
063 C*****C
064 C COMPLETE THE BOTTOM TRIANGLE OF THE SYMMETRIC MATRIX.            C
065 C*****C
066     DO 6 M=2,NUE
067         DO 7 N=1,M-1
068             H(M,N)=H(N,M)
069     7     CONTINUE
070     6     CONTINUE

```

```

671      RETURN
672      END
673 C
674      SUBROUTINE NORM22(N,A,AN)
675 C*****C
676 C THIS SUBROUTINE COMPUTES THE EUCLIDIAN NORM SQUARED OF A. C
677 C*****C
678      REAL*4 A(*)
679      AN=0.0
680      DO 1 I=1,N
681          AN=AN+A(I)*A(I)
682 1      CONTINUE
683      RETURN
684      END
685 C
686      SUBROUTINE MATVEC(MDIM,NUE,H,P,Q)
687 C*****C
688 C THIS SUBROUTINE COMPUTES THE EUCLIDIAN NORM SQUARED OF A. C
689 C*****C
690      REAL*4 H(MDIM,MDIM),P(*),Q(*)
691      DO 1 M=1,NUE
692          SUM=0.0
693          DO 2 N=1,NUE
694              SUM=SUM+H(M,N)*P(N)
695 2      CONTINUE
696          Q(M)=SUM
697 1      CONTINUE
698      RETURN
699      END
700 C
701      SUBROUTINE DEGREE(NUE,W,R)
702 C*****C
703 C THIS SUBROUTINE CONVERTS RADIAN INTO DEGREES C
704 C*****C
705      REAL*4 W(*),R(*)
706      PI=.31415926E+01
707      DO 1 I=1,NUE
708          R(I)=(180.0/PI)*W(I)
709 1      CONTINUE
710      DO 2 I=1,NUE
711 3      CONTINUE
712          IF(R(I) .LT. -180.0) THEN
713              R(I)=R(I)+360.0
714              GO TO 3
715          ELSE IF(R(I) .GT. 180.0) THEN
716              R(I)=R(I)-360.0
717              GO TO 3
718          ELSE

```

```

719      END IF
720      2  CONTINUE
721      RETURN
722      END
723  C
724      SUBROUTINE BEAMWD(NUE,X,Y,AI,W,HT,HPBW,BWFN)
725  C*****C
726  C THIS SUBROUTINE COMPUTES THE HALF POWER BEAMWIDTH AND THE BEAMWIDTH C
727  C BETWEEN FIRST NULLS OF THE POWER PATTERN. C
728  C*****C
729      REAL*4 X(*),Y(*),AI(*),W(*)
730      RAD=.17453293E-01
731      TP=.62831853E+01
732      TP2=.39478418E+02
733  C*****C
734  C COMPUTE THE POWER IN THE MAIN BEAM. C
735  C*****C
736      ARG1=HT*TP
737      S=SIN(ARG1)
738      BO=S*S/TP2
739      SUM1=0.0
740      SUM2=0.0
741      DO 4 I=1,NUE
742          SUM1=SUM1+AI(I)*COS(W(I))
743          SUM2=SUM2+AI(I)*SIN(W(I))
744      4  CONTINUE
745      PO=BO*(SUM1*SUM1+SUM2*SUM2)
746  C*****C
747  C FIND THE HALF POWER POINT AND THE FIRST NULL POWER POINT. C
748  C*****C
749      T=0.0
750      DT=.05
751      PLOWER=1.0
752      ITAG=0
753      P=45.0
754      RP=RAD*P
755      TPCP=TP*COS(RP)
756      TPSP=TP*SIN(RP)
757      5  CONTINUE
758          T=T+DT
759          R=TP*SIN(RAD*T)
760          ST=SIN(RAD*T)
761          U=ST*TPCP
762          V=ST*TPSP
763          U2=U*U
764          V2=V*V
765          ARG1=HT*SQRT(TP2-U2-V2)
766          ARG2=0.25*U

```

```

767      S=SIN(ARG1)
768      C=COS(ARG2)
769      BI=S*S*C*C/(TP2-U2)
770      S1=0.0
771      S2=0.0
772      DO 6 K=1,NUE
773          UX=U*X(K)
774          VY=V*Y(K)
775          PSI=COS(X(K)*U)*COS(Y(K)*V)
776          S1=S1+AI(K)*PSI*COS(W(K))
777          S2=S2+AI(K)*PSI*SIN(W(K))
778      6  CONTINUE
779      POI=BI*(S1*S1+S2*S2)/PO
780      IF(POI .LE. PLOWER) THEN
781          IF((POI .LE. 0.5).AND.(ITAG .EQ. 0)) THEN
782              HPBW=2.0*T
783              ITAG=1
784          ELSE
785              END IF
786          PLOWER=POI
787          GO TO 5
788      ELSE
789          BWFN=2.0*T
790          GO TO 7
791      END IF
792      7  CONTINUE
793      RETURN
794      END

```

## LABORATORY OPERATIONS

The Aerospace Corporation functions as an "architect-engineer" for national security projects, specializing in advanced military space systems. Providing research support, the corporation's Laboratory Operations conducts experimental and theoretical investigations that focus on the application of scientific and technical advances to such systems. Vital to the success of these investigations is the technical staff's wide-ranging expertise and its ability to stay current with new developments. This expertise is enhanced by a research program aimed at dealing with the many problems associated with rapidly evolving space systems. Contributing their capabilities to the research effort are these individual laboratories:

**Aerophysics Laboratory:** Launch vehicle and reentry fluid mechanics, heat transfer and flight dynamics; chemical and electric propulsion, propellant chemistry, chemical dynamics, environmental chemistry, trace detection; spacecraft structural mechanics, contamination, thermal and structural control; high temperature thermomechanics, gas kinetics and radiation; cw and pulsed chemical and excimer laser development, including chemical kinetics, spectroscopy, optical resonators, beam control, atmospheric propagation, laser effects and countermeasures.

**Chemistry and Physics Laboratory:** Atmospheric chemical reactions, atmospheric optics, light scattering, state-specific chemical reactions and radiative signatures of missile plumes, sensor out-of-field-of-view rejection, applied laser spectroscopy, laser chemistry, laser optoelectronics, solar cell physics, battery electrochemistry, space vacuum and radiation effects on materials, lubrication and surface phenomena, thermionic emission, photosensitive materials and detectors, atomic frequency standards, and environmental chemistry.

**Electronics Research Laboratory:** Microelectronics, solid-state device physics, compound semiconductors, radiation hardening; electro-optics, quantum electronics, solid-state lasers, optical propagation and communications; microwave semiconductor devices, microwave/millimeter wave measurements, diagnostics and radiometry, microwave/millimeter wave thermionic devices; atomic time and frequency standards; antennas, rf systems, electromagnetic propagation phenomena, space communication systems.

**Materials Sciences Laboratory:** Development of new materials: metals, alloys, ceramics, polymers and their composites, and new forms of carbon; nondestructive evaluation, component failure analysis and reliability; fracture mechanics and stress corrosion; analysis and evaluation of materials at cryogenic and elevated temperatures as well as in space and enemy-induced environments.

**Space Sciences Laboratory:** Magnetospheric, auroral and cosmic ray physics, wave-particle interactions, magnetospheric plasma waves; atmospheric and ionospheric physics, density and composition of the upper atmosphere, remote sensing using atmospheric radiation; solar physics, infrared astronomy, infrared signature analysis; effects of solar activity, magnetic storms and nuclear explosions on the earth's atmosphere, ionosphere and magnetosphere; effects of electromagnetic and particulate radiations on space systems; space instrumentation.